P.S.I.

PAR
JACQUES
DECONCHAT

P.S.I.

102 PROGRAMMES POUR

YENO SEGA 3000

Ouvrages d'initiation

— Visa pour l'informatique, Jean-Michel Jégo.
— Visa pour le Basic, Jean-Michel Jégo.
— Mon ordinateur, Jean-Claude Barbance.
— Le logotron informatique, Jean-Pierre Petit.
— L'ordinateur apprivoisé, François Picard et Danielle Shaw.
— Minitel mode d'emploi, Dominique Schraen et Sibylle Rochas.
— Macintosh le magnifique, Merl K. Miller et Mary A. Myers, traduit par Raymond Borraz.

La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'article 41, d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective », et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause, est illicite » (alinéa 1er de l'article 40).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles 425 et suivants du Code Pénal.

© *Editions du P.S.I. B.P. 86, 77402 LAGNY CEDEX, 1984*
ISBN : 2-86595-177-4

JACQUES
DECONCHAT

102 PROGRAMMES POUR
YENO
SEGA 3000

adapté par Gérard SANTRAILLE

EDITIONS DU P.S.I.
1984

SOMMAIRE

Avant-propos 8
Organisation de l'ouvrage 9

Niveau 1

Instructions élémentaires 13
Le nombre mystérieux 18
Cube 20
Chaud, froid 22
Les allumettes 24
Allumettes à plusieurs 26
Calendrier perpétuel 28
Roulette 30
Bataille navale 32
Golf 34
Solfège 36
Le jeu du 21 38
Pair - Impair 40
Simon 42
Devine un chiffre 44
Le jeu de Wytthof 46
421 48
Rebonds 50
Jackpot 52
Le jeu de la fève 54
Jeu de l'oie aléatoire 56

Niveau 2

Dimensionnement d'un tableau
Recherche d'un élément dans un tableau et dans une chaîne 60
Un dé 62
Visions 64
Plus grand, plus petit 66
L'électricien fou 68
La course au 2Ø 70
Dessin 72
Dé truqué 74
Le jeu de saute-mouton 76
Lettres 78

Espion 80
Accrochez les wagons 82
Tirage d'un dé truqué 84
Ordre 86
Course de chevaux 88
Le jeu de Marienbad 90
Idem 92
Mastermind 94
A travers 96
Reines sur l'échiquier 98
Le jeu de la vie 100

Niveau 3

Manipulation des chaînes et graphiques haute résolution 104
Losanges 106
Echiquier 108
Couloirs 110
Jour de la semaine 112
Le pendu 114
Bridge 116
Horloge 1 118
Le loup et les agneaux 120
Belote 122
Un taquin de calcul 124
Poker 126
Hanoï 128
Musique 130
Pendu inversé 132
Mathieu 134
Machinamot 136
Compose 138
Nicomaque 140
Permutations 142
Tour de cartes 144
Parcours du cavalier 146
Mur de briques 148

Niveau 4

Les techniques d'animation élémentaire 153
Tirage de dés 154
Jeu de loto 156
La bombe 158
Ardoise magique 160
Affichage publicitaire 162

Evasion 164
Chapeaux 166
Combat aérien 168
Dessin de dés 170
Le jeu du missile 172
Le singe 174
Le jeu du 15 176
Adresse 178
Sous-marin 180
Zombies 182
Diago 184
Topologie 186
Chasse au sous-marin 188
Le lézard d'Alice 190
Calculatrice 192
Smokey 194
Course de voitures 196
Echecs 198
Danger spatial 200
Tic tac toe 202
Gouttes 204
Obstacles 206
Captures dans l'espace 208
Plusieurs en un 210

Niveau 5

Présentation de certaines fonctions du Basic étendu 215
Horloge 2 216
Dames 218
Anagrammes 220
Œufs 222
Affichage tournant (vertical) 224
Quatre à quatre 226
Taquin de lettres 228
Cibles 230
Affichage publicitaire (2) 232
Anomalie 234
Othello 236

Avant-propos

Vous trouverez dans ce livre plus de 1ØØ programmes, rédigés en Basic niveau III A.

Le fait de ne présenter que des programmes très courts correspond ici à un choix : d'une part le désir de guider le lecteur dans un apprentissage progressif du langage, en utilisant pour cela des jeux courts, mais néanmoins assez complets et performants, d'autre part la volonté de ne pas tomber dans le piège classique de programmes trop longs, très difficiles à entrer en machine sans erreur, et qui sont dès lors plus à l'aise sur des cassettes préenregistrées que l'on trouvera par ailleurs de plus en plus couramment dans le commerce.

Chaque niveau de programme (il y en a 5) est précédé d'une courte étude des instructions Basic mises en œuvre dans les jeux qui suivent.

De plus, les programmes proposés sont "ouverts", ce qui signifie qu'ils ont été conçus, dans la mesure du possible, pour être facilement utilisés et modifiés.

Des indications sont fournies à la fin de chaque programme, pour vous suggérer des améliorations possibles, ou des modifications qui pourraient rendre le jeu plus attrayant, plus performant ou plus complet.

Organisation de l'ouvrage

Les jeux sont classés par difficulté croissante, suivant 5 niveaux :

- **Le niveau 1** permet d'étudier les instructions élémentaires du langage Basic : **LET, PRINT, INPUT, IF___THEN___, FOR...NEXT, GOTO, GOSUB, ON...GOTO, ON...GOSUB, CURSOR.**

- **Au niveau 2,** on utilisera les instructions de confection et de manipulation des éléments d'un tableau, et d'une chaîne : **DIM et MID$.**

- **Au niveau 3,** on fait appel à la puissante instruction de traitement des chaînes de caractères : **MID$** et aux instructions graphiques : **PATTERN, CURSOR** et **VPEEK.**

- **Au niveau 4,** on rencontre des instructions propres aux jeux intéractifs : **INKEY$.**

- Enfin, **le niveau 5** nous familiarisera avec certaines instructions du Basic Graphique, comme la manipulation des **SPRITES** (petits lutins animés) et avec l'affichage sur écran d'un caractère redéfini comme un sprite.

Chaque jeu est présenté d'une façon très systématique, par une courte description, suivie de l'étude détaillée de chacune des lignes importantes du programme. Après quoi, les indications nécessaires au bon fonctionnement du jeu sont données, ainsi que quelques éléments pour des extensions ou des améliorations possibles du jeu proposé.

Niveau 1

Nom du programme	*Thèmes et particularités*
Le nombre mystérieux	Jeu de déduction, pour un joueur. Plusieurs niveaux.
Cube	Jeu de dessin : perspective sur un écran.
Chaud, froid	Logique et hasard, pour un joueur. Plusieurs niveaux.
Les allumettes	Jeu de réflexion, contre le YENO.
Allumettes à plusieurs	Jeu de réflexion et de stratégie, pour plusieurs joueurs.
Calendrier perpétuel	Confection d'un calendrier, suivant mois et année.
Roulette	Jeu de hasard, pour plusieurs joueurs. Simulation.
Bataille navale	Un seul navire à trouver, mais quelques particularités.
Golf	Un parcours en 18 trous, pour un seul joueur.
Solfège	Apprenez le solfège avec l'aide de votre YENO.
Le jeu du 21	Jeu de hasard, contre le YENO.
Pair - Impair	Un jeu de casino en trois temps (un ou plusieurs joueurs).
Simon	Répétez une séquence de chiffres de plus en plus longue.
Devine un chiffre	Jeu de casino, pour un ou plusieurs joueurs.
Le jeu de Wytthof	Un jeu de réflexion très simple, pour deux joueurs.
421	Une simulation simplifiée, pour plusieurs joueurs.
Rebonds	Un jeu d'adresse très visuel, pour un joueur.
Jackpot	Jeu de hasard très classique (un joueur).
Le jeu de la fève	Un jeu de réflexion, pour deux joueurs.
Jeu de l'oie aléatoire	Un vieux jeu, à la sauce YENO pour deux joueurs.

Instructions élémentaires

Nous ne retiendrons ici que les instructions fondamentales, ou instructions de programmation ; beaucoup d'autres commandes, ou fonctions particulières, seront utilisées à l'occasion ; par exemple :

– ***CLS pour nettoyer l'écran.***
– ***RND pour obtenir un nombre aléatoire entre 0*** et 1.
– ***INT pour prendre la partie entière d'un résultat.***
etc.

Les instructions fondamentales de la programmation, utilisées dès le niveau 1, et présentes dans pratiquement tous les programmes, sont :
LET ; PRINT ; INPUT ; GOTO ; IF ... THEN ; FOR ... NEXT ; GOSUB ... (RETURN) ; CURSOR ;

soit seulement huit instructions de base que nous allons maintenant examiner en détail.

L'INSTRUCTION LET, OU INSTRUCTION D'AFFECTATION

C'est l'instruction par excellence des langages de programmation évolués, puisque son rôle essentiel est de permettre, pour le programmeur, une gestion transparente de la mémoire. Cela signifie, en pratique, que pour ranger une quantité en mémoire, par exemple 12, il n'est pas nécessaire de savoir dans quelle case de la mémoire on va la déposer : il suffit de lui affecter un nom autorisé, et c'est par ce nom que la quantité sera par la suite retrouvée et manipulée ;

– **si la quantité est numérique,** le nom sera une simple lettre de l'alphabet, ou un ensemble de plusieurs chiffres ou lettres, commençant obligatoirement par une lettre.

– **si la quantité est alphanumérique,** la même règle s'applique pour le nom, qui devra cependant être suivi du symbole $, et la variable devra être écrite entre guillemets.

Exemple : LET A1 = 12 ; la variable numérique A1 prend la valeur 12
LET NOM $ = "RENOIR" ; la variable alphanumérique NOM$ prend la valeur RENOIR.

Dans la plupart des Basic, l'instruction LET peut être omise, ce qui facilite l'écriture : A =12 ; B$ = "TOTO", etc.

L'affectation une fois réalisée, le micro-ordinateur saura, chaque fois qu'on lui parle de A1, qu'il s'agit du nombre 12 ; si on lui parle de NOM$, du mot RENOIR, etc.

L'INSTRUCTION PRINT, OU INSTRUCTION DE SORTIE (ECRAN)

C'est une instruction de dialogue, qui permet à la machine d'afficher sur l'écran un texte, le résultat d'un calcul, ou une variable ; c'est le seul moyen que possède le micro-ordinateur de faire connaître ses résultats. Ainsi, PRINT A1 provoque l'affichage de 12 (si l'on a fait auparavant LET A1 = 12), et PRINT NOM$ provoque l'affichage de RENOIR (si l'on a fait LET NOM$ = "RENOIR"). Utilisé après effacement d'écran (CLS), le PRINT provoque l'affichage sur la 1re ligne, en haut, à gauche.

Divers paramètres seront utilisés pour indiquer au YENO où il doit afficher ses résultats ou ses symboles (y compris les dessins).

- **CURSOR X, Y** prépare l'affichage sur la colonne X ($0 \leq X \leq 39$) à la ligne Y ($0 \leq Y \leq 23$).

- **PRINT TAB(N)** provoque sur la ligne d'impression courante un affichage commençant sur la Nième colonne à partir de la gauche ($0 \leq N \leq 39$).

- Un ; (**point virgule**) en fin de PRINT permet, soit d'enchaîner une autre impression sans répéter l'instruction PRINT, soit d'indiquer que la prochaine impression sera, en l'absence d'indication contraire, effectuée exactement à la suite de l'impression précédente.

- La , (**virgule**) provoque une tabulation automatique prédéfinie.

- Le : (**deux points**) permet de revenir à la ligne.

L'INSTRUCTION INPUT, OU INSTRUCTION D'ENTREE (CLAVIER)

Cette instruction permettra d'entrer et d'affecter directement dans la variable désignée par le INPUT un nombre, ou un texte, et ceci en mode conversationnel, c'est-à-dire au fur et à mesure des besoins ; en effet, lorsque le YENO rencontre une instruction INPUT, il s'arrête et vous rend la main ; vous pourrez alors entrer une information de nature convenable et lui signaler que cette entrée est terminée à l'aide de la touche **CR** qui rendra la main au YENO.

Par exemple, si dans le programme, le YENO trouve INPUT B, il s'arrête, et vous devez taper un nombre B quelconque, suivi de CR (si vous tapez un caractère autre qu'un chiffre ou un point décimal, vous aurez un message d'erreur : il est prudent de prévoir l'affichage d'un texte avant l'arrêt, vous précisant ce que vous devez entrer).

Par exemple : INPUT "TEXTE A AFFICHER" : B
Il existe d'autres moyens d'entrer des informations (LET, DATA, etc.), mais ils n'ont pas la souplesse de INPUT, puisque l'information est alors figée dans le programme, et n'est que difficilement modifiable.

L'INSTRUCTION GOTO OU INSTRUCTION DE RENVOI INCONDITIONNEL

Une des particularités du langage Basic, qui le rend facilement accessible, est que ***chaque ligne de programme doit être précédée d'un numéro,*** ces lignes étant ensuite exécutées par le micro-ordinateur en séquence, dans le sens des numéros croissants. Pour diverses raisons, il pourra être nécessaire, au cours du déroulement d'un programme, de ne pas suivre cet ordre séquentiel très rigide ; on utilisera alors ***l'instruction GOTO, qui permet de détourner le déroulement du programme de son cours normal,*** quitte d'ailleurs à le reprendre ensuite par un autre GOTO.

Le Basic, et c'est le cas du Basic YENO, n'autorise en général que les renvois à un numéro de ligne connu (GOTO 2Ø, par exemple, ou GOTO 2ØØ, etc.).

Mais une des particularités les plus remarquables du Basic YENO est qu'il permet aussi le GOTO calculé.

En effet, une expression du type **ON** expression numérique, **GOTO** ligne, ligne, ligne, s'exécute en deux temps. L'ordinateur évalue l'expression numérique, ou la variable, située entre **ON** et le **GOTO** qui est arrondie à un entier. Il se dirige ensuite vers le numéro de la ligne à droite du **GOTO** situé à la position pointée par la variable précédente.

Par exemple : ON A GOTO 1ØØØ, 2ØØØ, 5ØØ branchera le programme en 1ØØØ si A = 1, en 2ØØØ si A = 2 et en 5ØØ si A = 3. Notons qu'il n'y a pas de limite au nombre de numéros de lignes situé à droite du GOTO.

L'INSTRUCTION IF___THEN___ OU INSTRUCTION DE TEST

L'instruction **IF___THEN___** est très puissante. On l'emploie en rupture de séquence conditionnelle, sous la forme :

IF condition = vraie THEN nnn numéro de ligne.

C'est l'usage le plus courant de ce test, qui pourra toutefois être amélioré ici par le fait que la condition peut être une expression arithmétique ou logique complexe, évaluée logiquement.

L'INSTRUCTION FOR...NEXT, OU INSTRUCTION DE BOUCLES

On sera très souvent amené, en programmation, à vouloir exécuter un nombre entier de fois une certaine suite d'instructions (par exemple, faire avancer un mobile de la case Ø à la case 2Ø, etc.). Plutôt que d'utiliser un test et un renvoi, il sera préférable de faire appel à une ***boucle,*** la variable de boucle prenant ses valeurs dans les entiers ou les décimaux, et le ***pas*** de cette variable (quantité dont elle augmente ou diminue à chaque passage) pouvant aussi prendre ses valeurs dans les entiers ou les décimaux.

Ainsi, si l'on écrit :

```
FOR I =    Ø TO 2Ø STEP 2
  : \      Instructions
NEXT I
```

Les instructions contenues dans la boucle seront exécutées pour I = Ø, pour I = 2, ... et ceci jusqu'à I = 2Ø, soit 11 fois. La variable I peut être utilisée pour faire un calcul dans la boucle, mais ne devra pas être modifiée dans la boucle. On peut, bien sûr, sortir de la boucle, avant que I = 2Ø, par un test portant une quantité quelconque.

Si ***le pas de la boucle (STEP)*** n'est pas précisé, il sera automatiquement pris égal à 1.

Le Basic permet d'emboîter un nombre indéfini de boucles (limité par la mémoire disponible).

L'INSTRUCTION GOSUB, OU INSTRUCTION D'APPEL, DE SOUS-PROGRAMME

Lorsqu'une même suite d'instructions doit être exécutée plusieurs fois, et appelée à partir de plusieurs endroits du programme, plutôt que de la réécrire plusieurs fois, on essaiera de la rédiger une fois pour toutes, et on la placera en dehors du programme proprement dit (en lui donnant, par exemple, un numéro de ligne de départ supérieur au numéro de la dernière ligne du programme). Indépendamment des difficultés que l'on pourra rencontrer au niveau des noms des variables, lorsque l'on sera désireux d'utiliser cette séquence, il suffira de l'appeler par **l'instruction GOSUB numéro de ligne.** Le programme ira dès lors "faire un détour" dans la partie du programme appelée, mais il ne pourra revenir et poursuivre son déroulement normal que si il rencontre dans le programme appelé une instruction **RETURN,** ou instruction de retour.

L'INSTRUCTION CURSOR

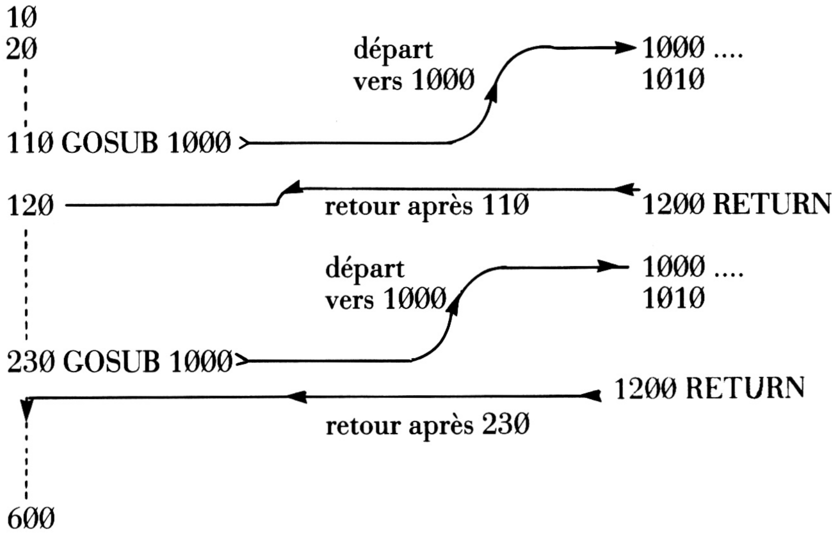
Cette instruction, largement employée dans cet ouvrage, permet d'afficher un caractère en un point quelconque de l'écran, en combinaison avec PRINT.

CURSOR n° de colonne, n° de ligne : PRINT (chaîne à écrire).

Exemple : CURSOR 18,12 : PRINT "YENO".

Ci-après, le schéma théorique de fonctionnement :

Programme principal



Fin du programme principal

1ØØØ

12ØØ RETURN

Sous-programme 1

Le sous-programme n'est écrit qu'une fois, mais appelé aussi souvent que l'on veut.

Un sous-programme pourra lui-même éventuellement appeler un autre sous-programme, et ainsi de suite, sans autre limitation que la taille mémoire, dans le Basic.

Il est important de préciser que, dans le Basic, comme pour le GOTO, le GOSUB peut être aussi calculé, ce qui permettra une très grande concision et une très grande souplesse de programmation.

Par exemple : ON A GOSUB 83Ø, 541Ø provoquera l'exécution du sous-programme 83Ø si A = 1, et du sous-programme 541Ø si A = 2.

Le nombre mystérieux

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu de déduction pour un joueur.
Difficulté du jeu : facile.

LE JEU :
On demande à l'ordinateur de fabriquer un nombre de n chiffres (avec n compris entre 1 et 7).
On devra ensuite essayer de le découvrir, avec le moins d'essais possible, en recevant les seuls renseignements PLUS GRAND ou PLUS PETIT.

LE PROGRAMME :
La ligne 4Ø permet de fabriquer le nombre à deviner.
Les entrées sont faites par les lignes 6Ø et 7Ø.
La ligne 8Ø teste si le résultat est exact.
La ligne 1ØØ signale un essai trop petit, et la ligne 12Ø un essai trop grand.
Le compteur d'essais est en ligne 13Ø.
Les lignes 15Ø à 18Ø s'occupent de l'affichage du résultat, et de la gestion de la partie suivante.

POUR JOUER :
On lancera le programme par RUN.
On obtient un premier affichage NOMBRE DE CHIFFRES ? auquel on répond par un entier entre 1 et 7, suivi de ENTER.
On obtient alors l'affichage ESSAI N° 1, suivi de la question VOTRE NOMBRE :.
On entre un nombre, et l'on presse CR.
On obtient alors une réponse TROP PETIT, ou TROP GRAND, ou encore, si l'on a trouvé juste un affichage de VOUS AVEZ TROUVEZ... EN... ESSAIS. Si le résultat n'est pas exact, on revient à ESSAI N° 2, et VOTRE NOMBRE.
En fin de partie, on obtient la question AUTRE PARTIE (O ou N) ? on répond O pour faire une autre partie suivi de CR.
On peut arrêter à tout moment en entrant une lettre en réponse à la question VOTRE NOMBRE.

EXTENSIONS POSSIBLES :
On pourra imaginer de faire deviner au joueur autre chose qu'un chiffre (par exemple, un mot, en utilisant l'ordre alphabétique, ou encore autre chose...).

LE NOMBRE MYSTERIEUX

```
10 CLS
20 INPUT "NOMBRE DE CHIFFRES :";N
30 IF N>7 THEN 10
40 X=INT(RND(1)*10^N+1)
50 J=1
60 PRINT "ESSAI NO ";J
70 INPUT "VOTRE NOMBRE :";M
80 IF M=X THEN 150
90 IF M>X THEN 120
100 PRINT "TROP PETIT"
110 GOTO 130
120 PRINT "TROP GRAND"
130 J=J+1
140 GOTO 60
150 CLS
160 PRINT "VOUS AVEZ TROUVE ";X;" EN "
;J;" COUPS "
170 INPUT "UNE AUTRE PARTIE (O/N) ?";R
$ •
180 IF R$="O" THEN 10
```

Cube

REFERENCES :
Difficulté de programmation : aucune.
Catégorie de jeu : dessin.
Difficulté du jeu : aucune.

LE JEU :
Le programme proposé se contente de réaliser à l'aide d'instructions CURSOR une vue d'un cube en perspective cavalière.
Il pourrait être, par exemple, incorporé à un jeu basé sur les capacités de reconnaissance des formes géométriques élémentaires.

LE PROGRAMME :
Le dessin du cube est réalisé par l'impression de points, en utilisant une boucle (100-150) pour l'impression des arêtes fuyantes.
Le nom de l'objet représenté est affiché en ligne 160 (ici : CUBE).

POUR JOUER :
Il suffit de lancer l'exécution par RUN, et de regarder.
Sortie du programme par BREAK en fin d'exécution.

EXTENSIONS POSSIBLES :
On pourra imaginer un jeu utilisant des parallélépipèdes rectangles de différentes tailles, avec affichage par exemple des 3 côtés, et il faudrait trouver leur volume, etc.

CUBE

```
10 CLS
20 FOR I=7 TO 17:CURSOR I,10:PRINT".":
NEXT I
30 FOR I=7 TO 17:CURSOR I,20:PRINT".":
NEXT I
40 FOR I=11 TO 20:CURSOR 7,I:PRINT".":
NEXT I
50 FOR I=11 TO 20:CURSOR 17,I:PRINT"."
:NEXT I
60 FOR I=14 TO 24:CURSOR I,3:PRINT".":
NEXT I
70 FOR I=14 TO 24:CURSOR I,13:PRINT"."
:NEXT I
80 FOR I=4 TO 13:CURSOR 14,I:PRINT".":
NEXT I
90 FOR I=4 TO 13:CURSOR 24,I:PRINT".":
NEXT I
100 FOR I=1 TO 6
110 CURSOR 7+I,10-I:PRINT"."
120 CURSOR 17+I,10-I:PRINT"."
130 CURSOR 7+I,20-I:PRINT"."
140 CURSOR 17+I,20-I:PRINT"."
150 NEXT I
160 CURSOR 11,23:PRINT"CUBE";
170 GOTO 160
```

Chaud, froid

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : solitaire.
Difficulté du jeu : moyenne.

LE JEU :
Un des plus vieux jeux du monde, toujours aussi pratiqué, mais cette fois, c'est l'ordinateur qui vous sert de partenaire : il va en effet cacher un objet à un certain emplacement d'une grille dont vous aurez vous-même choisi les dimensions. Pour le trouver, il ne vous donnera que quelques indications sous la forme TU BRULES, ou C'EST FROID, etc.
Vous devez réussir à découvrir l'objet caché, avec le moins d'essais possible (votre score sera affiché en fin de partie).

LE PROGRAMME :
Les lignes 3Ø et 4Ø définissent la position de l'objet.
La ligne 9Ø calcule la distance par rapport aux deux coordonnées que vous avez entrées à la ligne 7Ø.
La ligne 1ØØ regarde si vous avez gagné.
Les lignes 12Ø et 13Ø normalisent la distance (qui doit avoir un sens, quelles que soient les dimensions de la grille initiale).
La ligne 16Ø effectue le départ, s'il y a lieu, vers l'un des sous-programmes d'affichage.

POUR JOUER :
On lance le programme par l'ordre RUN, puis on introduit les dimensions de la grille (carrée) et l'on frappe CR.
Dès l'affichage suivant (DONNEZ X, Y), on entre la coordonnée X supposée, la coordonnée Y supposée suivie de CR.
On obtient en réponse un texte, qui est variable selon que l'on sera plus ou moins loin de l'objet cherché (cela va de COMPLETEMENT GELE à BRULANT). Lorsque l'on a trouvé, on obtient l'affichage du nombre d'essais effectués.
Sortie du programme avant la fin par BREAK.

EXTENSIONS POSSIBLES :
Si l'on dispose de la couleur, il sera intéressant de trouver un moyen plus visuel pour symboliser la distance, par exemple des pavés allant du rouge au bleu, aux emplacements essayés.
On peut aussi travailler dans un espace à trois dimensions, cacher plusieurs objets, etc.

CHAUD, FROID

```
10 S=0
20 INPUT "DIM. DE LA GRILLE : ";D
30 X=INT(RND(1)*D+1)
40 Y=INT(RND(1)*D+1)
50 CLS
60 PRINT:PRINT
70 INPUT "DONNEZ X,Y : ";A,B
80 S=S+1
90 T=ABS(X-A)+ABS(Y-B)
100 IF T=0 THEN 290
110 PRINT:PRINT"VOUS ETES ";
120 N=INT(SQR(D))/T+1
130 IF N<=5 THEN 150
140 N=5
150 IF T>SQR(D)+2 THEN 170
160 ON N GOTO 190,210,230,250,270
170 PRINT"COMPLETEMENT GELE"
180 GOTO 60
190 PRINT"A PEINE TIEDE"
200 GOTO 60
210 PRINT"TIEDE"
220 GOTO 60
230 PRINT"CHAUD"
240 GOTO 60
250 PRINT"TRES CHAUD"
260 GOTO 60
270 PRINT"BRULANT"
280 GOTO 60
290 CLS
300 SOUND 1,700,10
310 FOR I=1 TO 50:NEXT I
320 SOUND 1,500,10
330 FOR I=1 TO 50:NEXT I
340 SOUND 1,900,10
350 FOR I=1 TO 50:NEXT I
360 SOUND 1,700,10
370 FOR I=1 TO 50:NEXT I
380 SOUND 0
390 PRINT"VOUS AVEZ TROUVE EN ";S;" CO
UPS"
```

Les allumettes

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : contre le YENO.
Difficulté du jeu : moyenne.

LE JEU :
Le joueur devra en début de partie choisir une certaine quantité d'allumettes, ainsi que la prise maximum autorisée. Il décide ensuite s'il commence, ou s'il laisse commencer le YENO. Chacun prend à tour de rôle autant d'allumettes qu'il le désire, dans la limite du nombre maximum autorisé.
Celui qui prend la dernière allumette est considéré comme perdant.

LE PROGRAMME :
Le jeu du joueur commence en ligne 6Ø.
Celui du YENO commence en ligne 1ØØ.
La ligne 7Ø contrôle la légalité du coup du joueur.
La ligne 8Ø gère le changement de joueur.
Le coup du YENO est calculé dans une boucle non formelle, de la ligne 11Ø à la ligne 14Ø.
Le résultat est affiché par la ligne 21Ø.
Si la partie est terminée, la ligne 22Ø le détecte et envoie en ligne 24Ø, pour le message de fin.

POUR JOUER :
On commence l'exécution par RUN, et, à l'affichage du message TOTAL, on entre le nombre d'allumettes, suivi de CR. Puis on entre la prise maximum autorisée, suivie de CR. Si le joueur veut commencer, il frappe ensuite Ø ; sinon il frappe 1, puis CR.
Après chaque coup, le nombre d'allumettes restantes est affiché, et le YENO affiche A VOUS.
Celui qui prend la dernière allumette a perdu.

EXTENSIONS POSSIBLES :
On pourra bien sûr envisager un affichage graphique couleur plus spectaculaire, imaginer de laisser le hasard choisir le nombre d'allumettes et la prise maximum, etc.

LES ALLUMETTES

```
10 CLS
20 INPUT"TOTAL ";N
30 INPUT"PRISE MAX ";I
40 INPUT"1ER VOUS[0]/MOI[1] ";D
50 IF D=1 THEN 100
60 INPUT "A VOUS ";P
70 IF [P<>INT[P]] OR [P<1] OR [P>I] OR
 [P>N] THEN 60
80 D=1
90 GOTO 190
100 S=I+D
110 T=[N-S]/[I+D]
120 IF T=INT[T] THEN 150
130 S=S-D
140 GOTO 110
150 P=ABS[S-D]
160 IF P=1 THEN 180
170 P=D
180 D=0
190 PRINT P
200 N=N-P
210 PRINT "RESTE ";N
220 IF N=0 THEN 240
230 GOTO 50
240 CLS
250 PRINT "BRAVO"
260 IF D=0 THEN 280
270 PRINT "POUR MOI"
280 END
```

Allumettes à plusieurs

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : logique (plusieurs joueurs).
Difficulté du jeu : assez difficile.

LE JEU :
Dans ce jeu pour plusieurs joueurs, le meneur de jeu devra, en début de partie, choisir un nombre d'allumettes, et indiquer au YENO le nombre de joueurs.
A chaque coup, le joueur qui a le trait est autorisé à retirer du tas d'allumettes restantes, jusqu'au double de ce que vient de retirer le précédent joueur ; celui qui retire la dernière allumette a gagné (ou perdu, si l'on modifie légèrement les règles).

LE PROGRAMME :
Les lignes 10 à 40 permettent les différentes entrées.
La ligne 80 réalise les impressions.
La ligne 100 contrôle le coup de chaque joueur (par rapport au coup précédent).
La ligne 140 gère les tours (remise à 1 du numéro du joueur).
La ligne 130 réalise le test de fin, et la ligne 180 affiche le numéro du joueur gagnant (avec la règle prévue initialement).

POUR JOUER :
Après avoir lancé le programme par RUN, on entre le nombre d'allumettes initial, suivi de CR puis le nombre de joueurs (et CR).
Le premier joueur ne pourra enlever qu'une allumette, le deuxième pourra en enlever 1 ou 2 ; le suivant 1, 2 ou 1, 2, 3 ou 4 selon le coup précédent, et ainsi de suite ; le retrait maximum autorisé est à chaque fois affiché par le YENO.
Lorsque la dernière allumette a été prise, on obtient l'affichage LE JOUEUR... A GAGNE, et la partie est terminée.

EXTENSIONS POSSIBLES :
On peut convenir que celui qui prend la dernière allumette est éliminé du jeu, et recommencer la partie entre les joueurs restants qui seront éliminés les uns après les autres.
On peut aussi imaginer une représentation graphique des tas d'allumettes, tirer au hasard le nombre à enlever en début de partie, etc.

ALLUMETTES A PLUSIEURS

```
10 CLS
20 INPUT "NOMBRE D'ALLUMETTES ";N
30 PRINT
40 INPUT "NOMBRE DE JOUEURS ";J
50 F=1
60 K=J
70 CLS
80 PRINT "JOUEUR N0 ";J-K+1 :PRINT "NO
MBRE MAXIMUM A OTER ";F :PRINT "NOMBRE
 TOTAL ";N
90 INPUT "VOTRE COUP ";C
100 IF (C=0) OR (C>F) THEN 90
110 N=N-C
120 F=2*C
130 IF N=0 THEN 180
140 IF K-1>0 THEN 160
150 K=J+1
160 K=K-1
170 GOTO 70
180 PRINT:PRINT "LE JOUEUR ";J-K+1;" A
 GAGNE"
```

Calendrier perpétuel

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : gadget publicitaire.
Difficulté du jeu : aucune.

LE JEU :
Ce programme vous permettra de confectionner tous les ans votre calendrier ; vous pourrez ainsi vous en servir pour envoyer à vos amis un calendrier du mois de leur naissance, ou pour leur souhaiter une bonne fête, etc.

LE PROGRAMME :
Les lignes 2Ø et 3Ø vous permettront d'entrer les mois et l'année.
Les lignes 4Ø et 5Ø calculent le numéro du premier jour du mois.
La ligne 6Ø calcule le nombre de jours du mois (il n'est pas tenu compte ici des 29 février, pour simplifier).
La boucle A (8Ø-12Ø) réalise l'impression du calendrier proprement dit.

POUR JOUER :
Après avoir lancé le programme par l'instruction RUN, on devra introduire l'année, sous la forme, par exemple 1983, suivie de CR.
On obtient alors la question MOIS, à laquelle il convient de répondre par le numéro du mois, suivant l'ordre usuel (1 pour JANVIER, 2 pour FEVRIER, etc.).
On fait ensuite CR, et le calendrier est alors affiché sur l'écran.
Pour en garder une trace, si l'on dispose bien sûr d'une imprimante, il suffira de modifier les PRINT en conséquence.

EXTENSIONS POSSIBLES :
On pourra sans difficulté prévoir l'affichage des 29 février, prévoir une option permettant l'affichage pour une année complète, prévoir l'introduction du nom des mois en clair, etc.

CALENDRIER PERPETUEL

```
10 CLS
20 INPUT "ANNEE ";A
30 INPUT "MOIS ";M
40 D=INT(365.25*(A-M-3))+INT(30.6*(13*
M+37))-INT((INT((A-M-3)/100)-7)*.75)+1

50 D=D-7*INT(D/7)
60 M=31+((M=4)+(M=6)+(M=9)+(M=11))+(M=
2)*3
70 PRINT:PRINT" DIM LUN MAR MER JEU VE
N SAM"
80 FORA=1TOM
90 PRINTTAB(4*(D+A-1)+2);STR$(A);
100 IF D+A-7<>0 THEN120
110 D=D-7:PRINT
120 NEXTA
130 PRINT:PRINT:PRINT:PRINT
```

Roulette

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : hasard (plusieurs joueurs).
Difficulté du jeu : facile.

LE JEU :
Il est surtout spectaculaire par son aspect graphique : une boule, représentée par un carré noir, se déplace sur un terrain de jeu comportant des cases numérotées de 1 à 36, ainsi que trois cases supplémentaires, le ØØ, une case P (comme pair) et une case I (comme impair).
La boule parcourt au moins une fois chaque case, puis s'arrête au deuxième tour ; on peut ensuite la relancer pour une autre partie.

LE PROGRAMME :
La ligne 2Ø dessine le terrain de jeu, auquel les lignes 3Ø-5Ø rajoutent les trois symboles P, ØØ et I.
Les boucles L et C (7Ø-19Ø) réalisent le déplacement de la boule, l'arrêt étant contrôlé par la ligne 13Ø.
La ligne 17Ø réalise l'effacement.
La ligne 9Ø évite la position inexistante sous la colonne P, ØØ et I.
La ligne 6Ø tire la position d'arrêt.

POUR JOUER :
Il suffit de faire RUN, et de regarder.
Une fois la boule arrêtée, il suffira de frapper une touche quelconque (sauf BREAK) pour la relancer. Taper BREAK pour sortir du programme.

EXTENSIONS POSSIBLES :
On pourra essayer de se rapprocher davantage d'une véritable roulette, de différentes manières : mieux dessiner le terrain de jeu, introduire des couleurs, simuler le ralentissement de la boule, introduire les différentes possibilités de pari, et assurer la gestion des gains correspondants, tenir les comptes de plusieurs joueurs, et annoncer les résultats sous une forme plus usuelle.

ROULETTE

```
10 CLS
20 PRINT"2  4  6  8  10 12 14 16 18":C
URSOR 0,4:PRINT"20 22 24 26 28 30 32 3
4 36":CURSOR 0,8:PRINT"1  3  5  7  9
11 13 15 17":CURSOR 0,12:PRINT"19 21 2
3 25 27 29 31 33 35"
30 CURSOR 27,1:PRINT"P"
40 CURSOR 27,5:PRINT"00"
50 CURSOR 27,9 :PRINT"I"
60 X=37+INT(36*RND(1))
70 FOR L=2 TO 14 STEP 4
80 FOR C=0 TO 27 STEP 3
90 IF (L=14) AND (C=27) THEN 180
100 CURSOR C,L:PRINT CHR$(144)
110 BEEP
120 X=X-1
130 IF X<>0 THEN 170
140 S$=INKEY$:IF S$="" THEN 140
150 CURSOR C,L:PRINT CHR$(32)
160 GOTO 60
170 CURSOR C,L:PRINT CHR$(32)
180 NEXT C
190 NEXT L
200 GOTO 70
```

Bataille navale

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : réflexion.
Difficulté du jeu : facile.

LE JEU :
Une bataille navale a priori assez simple, avec un seul navire caché au hasard dans une grille de 1Ø cases sur 1Ø cases.
A chaque coup, on devra tirer en donnant les coordonnées H (horizontale) et V (verticale) de la case visée, et le YENO répond en affichant la distance séparant le navire invisible de la case visée. Mais, si l'on tire trop près, le navire risque de se recacher au hasard.

LE PROGRAMME :
Les boucles I (3Ø-9Ø) et J (4Ø-8Ø) dessinent le terrain de jeu.
Les lignes 11Ø et 12Ø définissent l'emplacement du navire.
Les coordonnées du tir sont entrées aux lignes 18Ø et 23Ø, et la ligne 25Ø affiche le coup sur l'écran. La distance est calculée en ligne 26Ø, et la ligne 27Ø regarde si l'on a ou non fait mouche.
La ligne 31Ø intervient pour déplacer le navire si les coups sont rapprochés.
La ligne 41Ø permet l'affichage de la distance.

POUR JOUER :
Le terrain de jeu est affiché après avoir tapé RUN. A l'affichage de H, on répond en donnant le numéro de la ligne horizontale sur laquelle on désire tirer, et, à la question V, on répond en donnant le numéro de la colonne verticale.
Si on a tiré trop près, on obtient l'affichage PFTT... qui signifie que le navire ennemi a changé d'emplacement.
Si on a tiré sur la bonne case, on obtient l'affichage du nombre total de tirs effectués pour gagner.

EXTENSIONS POSSIBLES :
Elles sont innombrables, puisqu'aussi bien la bataille navale est l'un des jeux les plus répandus qui soit. On peut introduire deux joueurs, jouer contre le YENO, mettre plusieurs navires, des sous-marins, etc.

BATAILLE NAVALE

```
10 CLS
20 A=0
30 FOR I=A TO 9
40 FOR J=A TO 9
50 CURSOR 9+2*J,4:PRINT CHR$(48+J)
60 CURSOR 7+A,5+I:PRINT CHR$(48+I)
70 CURSOR 2*J+9,5+I:PRINT CHR$(46)
80 NEXT J
90 NEXT I
100 S=1
110 X=INT(10*RND(1))
120 Y=INT(10*RND(1))
130 CURSOR 1,19:PRINT SPC(32)
140 A=19
150 B=8
160 T$="H= "
170 GOSUB 510
180 GOSUB 540
190 H=VAL(L$)
200 B=18
210 T$="V= "
220 GOSUB 510
230 GOSUB 540
240 V=VAL(L$)
250 CURSOR 9+2*V,5+H:PRINT CHR$(48+S)
260 D=INT(SQR((X-H)*(X-H)+(Y-V)*(Y-V))
)
270 IF D=0 THEN 480
280 S=S+1
290 IF S-10 <> 0 THEN 310
300 S=0
310 IF D <> 1 THEN 400
320 T$="PFFTT....."
330 A=22
340 B=13
350 GOSUB 510
360 FOR I=1 TO 1000
370 NEXT I
380 CURSOR 1,A:PRINT SPC(32)
390 GOTO 110
400 CURSOR 1,21:PRINT SPC(32)
410 T$="DISTANCE="+STR$(D)
420 A=21
430 B=7
440 GOSUB 510
450 FOR I=1 TO 500
460 NEXT I
470 GOTO 130
480 CLS
490 PRINT,S;" COUPS":PRINT:PRINT
:PRINT:PRINT:PRINT:PRINT:PRINT
500 END
510 REM ROUTINE D'AFFICHAGE
520 CURSOR B,A:PRINT T$
530 RETURN
540 REM ROUTINE DE SAISIE
550 BEEP
560 L$=INKEY$
570 IF L$="" THEN 560
580 BEEP
590 CURSOR B+2,A:PRINT L$
600 FOR P=1 TO 400
610 NEXT P
620 RETURN
```

Golf

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : solitaire.
Difficulté du jeu : assez facile.

LE JEU :
Il s'agit d'essayer de réaliser, avec le moins de coups possible, un parcours de golf de 18 trous. Pour cela, il vous faut frapper la balle avec une force plus ou moins grande, jusqu'à obtenir sa chute dans le trou.
Mais attention, il peut y avoir un certain aléa sur chaque coup, et vous serez amené souvent à revenir en arrière (il faudra alors faire précéder l'indication de la force par un signe -).

LE PROGRAMME :
La ligne 4Ø permet de définir la position du trou, et la ligne 6Ø dessine le terrain de jeu.
Le déplacement à réaliser est calculé en ligne 1ØØ.
La boucle I (12Ø-17Ø) réalise le déplacement de la balle.
La ligne 18Ø regarde si le coup est, ou non, gagnant ; s'il l'est, on recommence (ligne 23Ø) jusqu'à ce que l'on ait réalisé 18 trous (boucle S (2Ø-23Ø) ; fin de boucle en 25Ø, avec affichage de fin).

POUR JOUER :
On obtient l'affichage du terrain en frappant RUN ainsi que le dessin de la balle sur sa position de départ, et le texte : TROU 1. FORCE ; on doit alors répondre en donnant la force désirée (un nombre entre 1 et 1ØØ...).
Dès que l'on aura tapé CR, la balle commence à se déplacer ; elle peut alors soit s'arrêter avant le trou (auquel cas on recommence), soit tomber dedans (on passe alors au trou suivant), soit le dépasser : on devra alors frapper avec une force négative.
Le jeu se termine avec l'affichage 18 TROUS EN ... COUPS.

EXTENSIONS POSSIBLES :
Il sera possible d'étendre le jeu pour faire participer plusieurs joueurs, d'améliorer la qualité graphique, d'utiliser une représentation spatiale du terrain (se référer à certains jeux d'arcades), d'avoir un terrain plus accidenté, des obstacles, de changer de club...

GOLF

```
10 E=0
20 FOR S=1 TO 18
30 X=2
40 T=INT(22*RND(1)+9)
50 CLS
60 CURSOR 0,16:FOR I=1 TO 32:PRINT CHR
$(155);:NEXT I
70 CURSOR T,16:PRINT" "
80 CURSOR X,15:PRINT"0"
90 CURSOR 0,18:PRINT"  TROU";STR$(S);"
: FORCE  ";:INPUT F
100 F1=.16*F:F=(ABS(F)*(SQR(RND(1))+.1
)):F=F*F/100:F=INT(F1+F)
110 IF F=0 THEN 50
120 FOR I=SGN(F) TO F STEP SGN(F)
130 X=X+SGN(F)
140 IF (X>32) OR (X<2) THEN 170
150 CURSOR X-SGN(F),14:PRINT" "
160 CURSOR X,14:PRINT"0"
170 NEXT I
180 IF X=T THEN 210
190 E=E+1
200 GOTO 50
210 CURSOR X,14:PRINT" "
220 CURSOR X,15:PRINT" "
230 NEXT S
240 CLS
250 PRINT" 18 TROUS EN ";E;" COUPS"
```

Solfège

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion.
Difficulté du jeu : facile.

LE JEU :
Ce programme vous propose une initiation élémentaire au solfège, en vous permettant d'apprendre à reconnaître facilement les différentes notes, qui apparaissent les unes après les autres sur une portée ; après l'affichage de la note, vous devez entrer son nom en clair et votre score sera affiché en permanence.
Le programme est initialement prévu pour la clé de sol, mais il sera très facilement adaptable à toute autre clé.

LE PROGRAMME :
La boucle I (4Ø-6Ø) dessine la portée.
La note est choisie en ligne 7Ø et affichée en lignes 8Ø, 9Ø et 1ØØ.
La ligne 11Ø affiche le score.
La ligne 12Ø affiche le nom de la clé (ici clé de sol) et vous interroge.
La ligne 13Ø s'occupe de la fin de la partie.
La ligne 14Ø effectue le décodage de la note entrée.
La ligne 16Ø contrôle le résultat.

POUR JOUER :
Après avoir lancé le programme par RUN, la portée sera affichée, ainsi que la première note.
Vous répondrez à la question NOTE ? en donnant le nom en clair de la note (do, ré, mi, fa, sol, la, si), suivi de CR.
Le score vous indiquera si c'est exact ou non, et une nouvelle note sera affichée.
Pour arrêter, il suffira de répondre Ø suivi de CR, à la question NOTE ?.

EXTENSIONS POSSIBLES :
On pourra tout d'abord facilement modifier le programme pour lui permettre de vous faire travailler les différentes clés. On pourra aussi améliorer le graphisme, introduire la couleur, le son (lorsque l'on a répondu juste, on entendra la note, par exemple). On pourra aussi réaliser suivant le même principe un jeu vous permettant de reconnaître des accords, etc.

SOLFEGE

```
10 S=0
20 P=S
30 CLS
40 FOR I=1 TO 5
50 CURSOR 1,1+2*I:FOR J=1 TO 32:PRINT
CHR$(233);:NEXT J
60 NEXT I
70 X=INT(12*RND(1))
80 CURSOR 9,X:PRINT CHR$(227)
90 CURSOR 9,X+1:PRINT CHR$(227)
100 CURSOR 9,X+2:PRINT CHR$(236)
110 CURSOR 0,15:PRINT"SCORE ";S;" SUR
";P;" ESSAIS":PRINT:PRINT:PRINT
120 INPUT "CLE DE SOL, NOTE ?";R$
130 IF R$="0" THEN 190
140 Y=-1*(R$="FA")+2*(R$="MI")+3*(R$="
RE")+4*(R$="DO")+5*(R$="SI")+6*(R$="LA
")
150 P=P+1
160 IF (Y-1)*(Y-X+7) THEN 180
170 S=S+1
180 GOTO 30
190 END
```

Le jeu du 21

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu contre le YENO.
Difficulté du jeu : facile.

LE JEU :
On devra entrer une mise au début du jeu, et on pourra lancer les dés en tapant 1 ; à chaque lancer, on obtiendra un nouveau total, qui augmentera peu à peu, et se rapprochera de 21. On devra essayer de se rapprocher le plus possible de 21, sans dépasser ce nombre. Lorsque l'on décidera de s'arrêter, il suffira de taper Ø pour laisser jouer le YENO qui essaiera alors d'avoir un meilleur score.
A la fin de chaque partie, le gain sera affiché ainsi que le total des gains (ou des pertes).

LE PROGRAMME :
Les lignes 6Ø, 7Ø et 8Ø réalisent le contrôle de celui qui doit jouer (joueur ou YENO).
Le sous-programme (34Ø-35Ø) lance les dés.
Les lignes 1ØØ et 11Ø calculent le total fait par le joueur et l'affichent.
Le total du YENO est calculé en ligne 17Ø et affiché par la ligne 18Ø.
La ligne 21Ø contrôle la fin de jeu pour le YENO, et la ligne 25Ø cherche quel est le gagnant.
Le gain (ou la perte) sera affiché par la ligne 27Ø, et le total par la ligne 29Ø.
La ligne 31Ø contrôle la fin de partie.

POUR JOUER :
Taper RUN pour lancer le jeu.
A la question VOTRE MISE, entrer la mise initiale, suivie de CR, et taper 1 pour lancer le dé. Le premier tirage est affiché sous la forme, VOUS AVEZ... et on devra refaire 1 suivi de CR, tant que l'on désire recommencer à tirer.
Dès que l'on a terminé, on tape Ø et CR pour laisser jouer le YENO (qui tient ici le rôle de la banque).
Le gain (ou la perte, s'il y a le signe -) sera affiché en fin de jeu, ainsi que le total des gains (ou pertes) réalisé depuis le début. Si l'on désire arrêter de jouer, on devra taper N en réponse à la question SUITE (O ou N) ?

EXTENSIONS POSSIBLES :
Ce jeu sera évidemment plus intéressant s'il est pratiqué à plusieurs, et il sera aisé de modifier le programme en conséquence ; on pourra aussi dessiner le dé, ou mieux, utiliser des cartes, ce qui permettra alors de faire un jeu de Black Jack.

LE JEU DU 21

```
20 G=0
30 CLS
40 INPUT "VOTRE MISE ";M
50 T=0
60 PRINT :PRINT :PRINT "1 POUR TIRER,
0 POUR FINIR ":PRINT :PRINT
70 INPUT R
80 IF R=0 THEN 130
90 GOSUB 340
100 T=T+X
110 PRINT :PRINT :PRINT "TOTAL :";T
120 GOTO 70
130 CLS
140 PRINT "LA BANQUE JOUE "
150 S=0
160 GOSUB 340
170 S=S+X
180 PRINT :PRINT :PRINT "TOTAL ";S
190 FOR I=1 TO 500
200 NEXT I
210 IF (S>T) OR (S>=17) THEN 230
220 GOTO 160
230 CLS
240 PRINT :PRINT :PRINT "VOUS AVEZ ";T
:PRINT :PRINT "LA BANQUE A ";S
250 IF (S>21) OR (S<T) AND (T<=21) THE
N 270
260 M=-M
270 PRINT :PRINT :PRINT "GAIN :";M
280 G=G+M
290 PRINT :PRINT :PRINT "TOTAL :";G
300 INPUT "SUITE (O/N) ?";R$
310 IF R$="N" THEN 330
320 GOTO 30
330 END
340 X=INT(RND(1)*6+1)
350 RETURN
```

Pair-impair

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : facile.

LE JEU :
C'est un jeu de hasard, pour un ou plusieurs joueurs. Le YENO lance simultanément deux dés, non visibles et pose trois questions : quelle est la parité du total, quel est le total, et enfin donner la valeur de l'un des dés. On peut ensuite soit poursuivre le jeu, soit l'interrompre à tout moment.

LE PROGRAMME :
Les dés sont tirés par un sous-programme placé aux lignes 360 et 370.
La ligne 70 tient le compte du nombre de parties, et les lignes 160, 240 et 280 comptabilisent les points marqués.
La parité du total est déterminée par les lignes 110 et 120 (la ligne 120 compare avec la réponse du joueur).
La ligne 310 réalise l'affichage de fin de partie : total, nombre de points, nombre de parties jouées.
La ligne 330 provoque l'arrêt si l'on répond N à la question ON CONTINUE.

POUR JOUER :
On lance le programme par l'instruction RUN.
Il suffit ensuite de répondre aux diverses questions posées, en suivant scrupuleusement les indications données et en tapant CR après chaque réponse.
Lorsque l'on a terminé une partie, on obtient l'affichage du total actuel et du nombre de parties sous la forme de, par exemple, 6 SUR 10.
On obtient alors l'affichage de la question ON CONTINUE (O ou N) et l'on devra répondre N suivi de CR pour arrêter.

EXTENSIONS POSSIBLES :
Si l'on désire jouer à plusieurs, il sera intéressant de prévoir la possibilité de miser, et d'ajouter une gestion du score réalisé par chacun des joueurs. On pourra aussi prévoir une amélioration sur le plan de la présentation (utilisation de la couleur, etc.).

PAIR-IMPAIR

```
20 T=0
30 R=T
40 GOSUB 360
50 A=X
60 GOSUB 360
70 R=R+1
80 S=A+X
90 CLS
100 INPUT "TOTAL PAIR (0) OU IMPAIR (1
) ";N
110 P=S/2-INT(S/2)
120 IF (N=P) OR (N-.5=P) THEN 150
130 PRINT :PRINT :PRINT "FAUX"
140 GOTO 170
150 PRINT :PRINT :PRINT "VRAI"
160 T=T+1
170 PRINT :PRINT
180 INPUT "VOTRE TOTAL :";N
190 IF N=S THEN 240
200 PRINT :PRINT :PRINT "PERDU"
210 FOR P=1 TO 300
220 NEXT P
230 GOTO 300
240 T=T+2
250 PRINT :PRINT
260 INPUT "BRAVO, DONNEZ UN DE ?";N
270 IF (N<>A) AND (N<>S-A) THEN 200
280 T=T+3
290 GOTO 210
300 CLS
310 PRINT "TOTAL :";T;"/";R:PRINT :PRI
NT
320 INPUT "ON CONTINUE ? (O/N) ";R$
330 IF R$="O" THEN 40
340 END
360 X=INT(RND(1)*6+1)
370 RETURN
```

Simon

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu de mémoire (1 ou plusieurs joueurs).
Difficulté du jeu : moyenne.

LE JEU :
Des nombres vont apparaître un par un.
Le joueur devra à chaque coup les répéter tous, depuis le début (ce qui signifie qu'il aura un de plus à taper à chaque fois).
Il devra essayer de tenir le plus longtemps possible. Le meilleur score est mémorisé après chaque partie, ce qui permettra éventuellement de jouer à plusieurs.

LE PROGRAMME :
La ligne 3Ø appelle un sous-programme (lignes 27Ø à 29Ø) qui donne la règle du jeu.
La ligne 7Ø fabrique les chiffres, qui sont ensuite stockés dans une variable numérique A (ligne 8Ø).
La ligne 9Ø permet l'impression du dernier chiffre au hasard sur la sixième ligne de l'écran, et la durée de l'affichage est déterminée par la ligne 11Ø.
Les lignes 14Ø et 15Ø attendent la réponse du joueur, qui est comparée à la réponse correcte en ligne 16Ø.
La ligne 18Ø marque les points.
La ligne 19Ø contrôle le meilleur score.
Les affichages de fin de partie sont effectués de la ligne 22Ø à la ligne 26Ø.

POUR JOUER :
Le lancement du programme par RUN provoque l'affichage de la règle du jeu, puis d'un chiffre qui disparaît. L'appareil pose alors la question VOTRE SEQUENCE ?
On lui répond en redonnant le chiffre, suivi de CR. Si c'est exact, on a un court instant l'affichage de EXACT. ATTENTION, puis un nouveau chiffre est affiché, qui s'efface et de nouveau la question VOTRE SEQUENCE ?
On retape alors deux chiffres déjà obtenus, suivis de CR.
On continue ainsi (3 chiffres, puis 4, 5, etc.), jusqu'à ce que l'on fasse une erreur, ce qui provoque l'affichage du score (et du meilleur score obtenu jusqu'à présent), suivi de la question AUTRE PARTIE (O ou N) ? ; on tape O pour continuer.
On peut aussi quitter le programme à tout moment, en tapant BREAK.

EXTENSIONS POSSIBLES :
On pourra réaliser des jeux analogues et plus ou moins perfectionnés en remplaçant les chiffres par des lettres, ou des couleurs, ou des sons, ...

SIMON

```
10 M=0
20 A=0
30 GOSUB 270
40 H=0
50 FOR I=1 TO 300
60 NEXT I
70 X=INT(10*RND(1))
80 A=10*A+X
90 PRINT TAB(28*RND(1)+4);X
100 BEEP
110 FOR I=1 TO 300
120 NEXT I
130 GOSUB 270
140 PRINT"VOTRE SEQUENCE ?"
150 INPUT R
160 IF R<>A THEN 220
170 PRINT"EXACT . ATTENTION"
180 H=H+1
190 IF M>=H THEN 50
200 M=H
210 GOTO 50
220 CLS
230 PRINT"VOTRE SCORE ";H:PRINT"MEILLE
UR SCORE ";M
240 INPUT "AUTRE PARTIE (O/N) ? ";R$
250 IF R$="N" THEN 300
260 GOTO 20
270 CLS
280 PRINT"VOUS DEVEZ REPETER TOUS LES
NOMBRES (UN DE PLUS A CHAQUE FOIS)"
290 RETURN
300 END
```

Devine un chiffre

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : facile.

LE JEU :
C'est un jeu de hasard, de conception très simple, mais néanmoins assez passionnant : le joueur introduit sa mise, et parie sur un chiffre, de 1 à 6 ; trois dés sont alors lancés ; si le chiffre joué apparaît une fois, le joueur gagne sa mise. S'il apparaît deux fois il double sa mise ; s'il apparaît trois fois, il triple sa mise.

LE PROGRAMME :
Le tirage des dés est réalisé par un sous-programme placé en lignes 26Ø-27Ø.
L'affichage est réalisé par la ligne 12Ø.
La ligne 14Ø calcule le gain dans les diverses configurations possibles.
La totalisation est effectuée en ligne 17Ø pour les gains, et en ligne 2ØØ pour les pertes.
La variable R$, testée en ligne 23Ø, détermine la poursuite éventuelle du jeu.

POUR JOUER :
Le lancement du programme par RUN provoque un premier affichage VOTRE MISE ? ; on répond par un nombre suivi de CR. On obtient alors un deuxième affichage UN CHIFFRE DE 1 A 6 ? ; on entre un chiffre suivi de CR ; les 3 dés, repérés par les lettres A, B, C s'affichent alors, ainsi qu'un message indiquant votre gain ou votre perte.
Puis une indication sur votre total actuel apparaît, ainsi que la question ON CONTINUE (O ou N) ? ; taper O pour continuer, N pour arrêter (suivi de CR).

EXTENSIONS POSSIBLES :
On pourra facilement introduire des contrôles (mise minimum, coup proposé de 1 à 6, etc.). On pourra aussi dessiner des dés avec un graphisme plus élaboré, introduire la possibilité de jouer à plusieurs, faire même participer le YENO.

DEVINE UN CHIFFRE

```
10 G=0
20 CLS
30 INPUT "VOTRE MISE : ";N
40 PRINT :PRINT
50 INPUT "UN CHIFFRE DE 1 A 6 :";D
60 GOSUB 260
70 A=X
80 GOSUB 260
90 B=X
100 GOSUB 260
110 C=X
120 PRINT :PRINT :PRINT TAB(4);A;TAB(8
);B;TAB(12);C:PRINT :PRINT TAB(5);"A";
TAB(9);"B";TAB(13);"C"
130 T=0
140 T=T-N*((D=A)+(D=B)+(D=C))
150 IF T=0 THEN 190
160 PRINT :PRINT "VOUS GAGNEZ ";T
170 G=G+T
180 GOTO 210
190 PRINT :PRINT "PERDU"
200 G=G-N
210 PRINT :PRINT "TOTAL ACTUEL : ";G
220 INPUT "ON CONTINUE (O/N) ?";R$
230 IF R$="N" THEN 250
240 GOTO 20
250 END
260 X=INT(RND(1)*6+1)
270 RETURN
```

Le jeu de Wytthof

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion (deux joueurs).
Difficulté du jeu : moyenne.

LE JEU :
On dispose d'une grille carrée de 1Ø cases sur 1Ø cases. Un point apparaît, en dehors de la diagonale et les joueurs devront le déplacer à chaque tour, suivant seulement trois directions : droite, bas et diagonale bas et droite. Le premier joueur qui arrive en (9,9) sera considéré comme gagnant, ou perdant, selon la convention faite au début.

LE PROGRAMME :
Les lignes 2Ø, 3Ø et 4Ø permettent de définir une position de départ, en dehors de la diagonale.
La boucle I (6Ø-1ØØ) dessine le terrain de jeu.
La ligne 14Ø imprime la position du point mobile.
Le test de fin est effectué en ligne 15Ø.
Le changement de joueur est assuré par les lignes 16Ø et 18Ø selon la valeur de H.
Les directions de déplacement disponibles, 1, 2 et 3, sont décodées par trois sous-programmes qui commencent aux lignes 26Ø, 28Ø et 29Ø.

POUR JOUER :
Taper RUN pour lancer l'exécution du programme. Le terrain de jeu s'affiche, ainsi que la position de départ.
Puis on obtient l'affichage JOUEUR N° 1, 2, 3 ? auquel on répond en tapant l'un de ces trois nombres.
C'est ensuite au tour du joueur 2.
Le jeu se termine lorsque l'un des joueurs arrive en case (9, 9).

EXTENSIONS POSSIBLES :
On peut tout d'abord améliorer les contrôles (la sortie du terrain n'est pas interdite, par exemple), améliorer ensuite l'affichage (graphisme, couleur), étendre le jeu à plusieurs joueurs avec un terrain agrandi, introduire un monstre qui va essayer de capturer un des joueurs, prévoir un programme qui permettrait au YENO de jouer, etc.

LE JEU DE WYTTHOFF

```
10 CLS
20 X=INT(5*RND(1))
30 Y=INT(5*RND(1))
40 IF X=Y THEN 30
50 A=0
60 FOR I=0 TO 9
70 CURSOR 13+I,3:PRINT CHR$(48+I)
80 CURSOR 12,4+I:PRINT CHR$(48+I)
90 CURSOR 13,4+I:PRINT ".........."
100 NEXT I
110 H=1
120 T$="JOUEUR N0    1,2,3 ?"
130 CURSOR 7,20:PRINT T$
140 CURSOR 13+Y,4+X:PRINT CHR$(144)
150 IF X+Y=18 THEN 250
160 CURSOR 17,20:PRINT"2"
170 IF H-1<>0 THEN 190
180 CURSOR 17,20:PRINT"1"
190 K$=INKEY$
200 IF K$="" THEN 190
210 BEEP
220 ON ASC(K$)-48 GOSUB 260,280,290
230 H=-H
240 GOTO 140
250 STOP
260 X=X+1
270 RETURN
280 GOSUB 260
290 Y=Y+1
300 RETURN
```

421

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : moyenne.

LE JEU :
Chacun des joueurs devra, à son tour, lancer 3 dés, et peut ensuite choisir, sur les deux coups suivants, s'il désire relancer ou non 1, 2 ou 3 dés. Il marque alors un score correspondant au nombre indiqué par les dés, suivant les règles choisies, et le joueur suivant procède de la même manière.

LE PROGRAMME :
Le nombre de joueurs est tout d'abord entré en ligne 1Ø.
La ligne 9Ø affiche le numéro de joueur et le nombre d'essais déjà effectués.
Un sous-programme placé en ligne 27Ø, utilisé 3 fois dans la boucle J (13Ø-16Ø), réalise le lancer de dés.
Les retirages sont effectués par les lignes 21Ø, 23Ø et 25Ø.
La variable R$ gère les 2ᵉ et 3ᵉ tirages. L'impression du résultat est réalisée dans tous les cas par la ligne 9Ø.
Les tests pour le nombre d'essais et pour le changement de joueur sont effectués aux lignes 17Ø et 19Ø.

POUR JOUER :
On lance l'exécution du programme par RUN ce qui conduit à l'affichage NBRE DE JOUEURS. On entre le nombre de joueurs, et l'on tape CR.
L'écran s'efface et affiche JR N° 1. ESSAI A, ainsi que le tirage réalisé, et les noms des dés (A, B ou C).
Si l'on ne désire pas refaire de tirage, il suffit de frapper CR. La main passe alors au joueur suivant.
Pour effectuer un autre tirage, il suffit de taper le nom du ou des dés à retirer, et de faire CR : par exemple, pour retirer A et C on tapera AC suivi de CR.
Les joueurs suivants procéderont de la même façon.

EXTENSIONS POSSIBLES :
La plus évidente consiste à assurer la gestion des résultats des joueurs, en se référant aux règles de gain usuelles du jeu du 421. On pourra envisager l'affichage après chaque tour d'un tableau récapitulant les résultats des joueurs.
On pourra aussi améliorer l'affichage des dés (se référer aux divers programmes d'affichage des dés se trouvant dans le même ouvrage).

421

```
10 INPUT "NOMBRE DE JOUEUR ";N
20 S=1
30 I=S
40 K=0
50 CLS
60 R$="ABC"
70 GOTO 130
80 K=K+S
90 CLS:PRINT,"421":PRINT,"---":PRINT:P
RINT"JR N0. ";I;" ESSAI";K:PRINT:PRINT
 TAB(4);A;TAB(8);B;TAB(12);C:PRINT:PRI
NT TAB(5);"A";TAB(9);"B";TAB(13);"C":P
RINT:PRINT:PRINT
100 INPUT "ABC ? ";R$
110 IF R$="" THEN 180
120 R$=R$+"DDD"
130 FOR J=S TO 3
140 GOSUB 270
150 ON ASC(MID$(R$,J,1))-64 GOSUB 210,
230,250,270
160 NEXT J
170 IF K<3 THEN 80
180 I=I+S
190 IF I<=N THEN 40
200 GOTO 30
210 A=D
220 RETURN
230 B=D
240 RETURN
250 C=D
260 RETURN
270 D=INT(RND(1)*5+S)
280 RETURN
```

Rebonds

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu d'adresse.
Difficulté du jeu : moyenne.

LE JEU :
Il utilise l'idée de la simulation d'un billard à bandes.
De part et d'autre de l'écran se trouvent la balle et le but visé : une série de chiffres et de lettres, en haut de l'écran, permettent de choisir l'endroit où l'on désire rebondir.
La balle va alors partir vers le haut de l'écran (vers le bas si l'on introduit un signe moins), y rebondir, revenir éventuellement frapper le bas, rebondir, et ainsi de suite jusqu'à atteindre la ligne de but ; si l'on est suffisamment près de la cible on a gagné et le nombre de rebonds sera alors affiché.

LE PROGRAMME :
Les boucles I (2Ø-4Ø) et (5Ø-7Ø) impriment les chiffres et les lettres en haut de l'écran.
Les lignes 8Ø et 9Ø définissent les positions initiales de la balle et de la cible.
Les lignes 1ØØ-11Ø effectuent leur impression.
Le symbole correspondant à la direction choisie est entré en ligne 12Ø.
La boucle I (15Ø-19Ø) réalise le déplacement de la balle.
La ligne 18Ø affiche la nouvelle position.
Le changement de direction, lors de la rencontre d'un bord, est effectué à la ligne 17Ø, par renvoi dans une partie du programme indépendante, située aux lignes 25Ø à 27Ø.
La ligne 21Ø effectue l'impression de fin de partie.
Le nombre de rebonds est compté en ligne 26Ø.

POUR JOUER :
On lance le programme par RUN.
Le terrain de jeu est affiché, ainsi que les positions initiales de la balle et de la cible. On choisit l'endroit où l'on désire faire rebondir la balle (de 1 à U), et on l'entre suivi de CR.
La trajectoire est alors affichée, suivie s'il y a lieu de la mention BRAVO ; X REBONDS.
On peut lancer une autre partie en tapant une touche quelconque, et s'arrêter à tout moment en tapant BREAK.

EXTENSIONS POSSIBLES :

Il sera très intéressant, sur un appareil disposant d'un graphisme plus fin, de dessiner un véritable billard par un segment, symbolisant la queue de billard, se déplaçant autour de la balle.

On peut aussi ajouter des trous, mettre plusieurs billes, sonoriser les rebonds, ou les heurts avec d'autres balles, les comptabiliser pour faire une partie à plusieurs joueurs, etc.

```
              REBONDS

10 CLS
20 FOR I=2 TO 6
30 CURSOR I,5:PRINT CHR$(51+I)
40 NEXT I
50 FOR I=7 TO 32
60 CURSOR I,5:PRINT CHR$(58+I)
70 NEXT I
80 X=INT(15*RND(1)+6)
90 Y=INT(15*RND(1)+6)
100 CURSOR 2,X:PRINT"0"
110 CURSOR32,Y:PRINT CHR$(144)
120 INPUT R$
130 S=0
140 T=X
150 FOR I=2 TO 32
160 T=T-X/(ASC(R$)-48+7*(R$)="A"))
170 IF(T>=22) OR (T<=5) THEN 250
180 CURSOR I,T:PRINT"0"
190 NEXT I
200 IF ABS(T-Y)>=3 THEN 220
210 PRINT "BRAVO:";S;" REBONDS"
220 F$=INKEY$
230 IF F$="" THEN 220
240 GOTO 10
250 X=-X
260 S=S+1
270 GOTO 190
```

Jackpot

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : hasard (un joueur).
Difficulté du jeu : facile.

LE JEU :
C'est une variante simplifiée d'un jeu de hasard classique, dans lequel le joueur doit faire une mise et tirer une poignée pour lancer le jeu. S'il obtient certaines combinaisons de symboles, il gagne (plus ou moins, selon sa mise et la combinaison obtenue).
Ici, les symboles sont remplacés par des chiffres, et le fait d'avoir trois chiffres identiques permet de gagner. Il existe de plus une combinaison particulière qui permet de remporter le Jackpot.

LE PROGRAMME :
La boucle I (3Ø-5Ø) permet le dessin de la machine, les lignes 6Ø, 7Ø et 8Ø affichent les emplacements (vides) des chiffres.
La mise sera entrée en ligne 9Ø, et testée en ligne 1ØØ : si on mise Ø, la partie est terminée (départ vers la ligne 35Ø).
Les lignes 11Ø, 13Ø et 15Ø permettent le tirage des 3 chiffres, en appelant un sous-programme qui se trouve en ligne 33Ø.
Le résultat est affiché par les lignes 16Ø, 18Ø et 2ØØ, puis testé en ligne 22Ø, et en ligne 3ØØ, si l'on a gagné.

POUR JOUER :
Lancer le programme par RUN ; à l'affichage de la question POT DE DEBUT, on répond en entrant un nombre suffisamment élevé, sur lequel vos mises seront décomptées.
Puis on tape CR, et on entre une première mise (différente de Ø), suivie de CR.
Les trois chiffres tirés sont alors affichés un bref instant, ainsi que la mise faite, et le résultat obtenu (gain ou perte).
On peut ensuite miser de nouveau, faire CR et ainsi de suite.
Pour terminer, il suffit de répondre Ø à la question VOTRE MISE.
Le restant de votre pot initial sera alors affiché.

EXTENSIONS POSSIBLES :
On pourra envisager d'améliorer sérieusement les contrôles, utiliser la couleur et des possibilités graphiques plus élaborées pour simuler davantage un véritable Jackpot.

JACKPOT

```
10 INPUT "POT DE DEBUT ";P
20 CLS
30 FOR I=1 TO 3
40 CURSOR 11,10+I:FOR J=1 TO 9:PRINT C
HR$(229);:NEXT J
50 NEXT I
60 CURSOR 12,12:PRINT" "
70 CURSOR 15,12:PRINT" "
80 CURSOR 18,12:PRINT" "
90 CURSOR 0,20:INPUT " VOTRE MISE ";M
100 IF M=0 THEN 350
110 GOSUB 330
120 A=X
130 GOSUB 330
140 B=X
150 GOSUB 330
160 CURSOR 12,12:PRINT CHR$(48+A)
170 BEEP
180 CURSOR 15,12:PRINT CHR$(48+B)
190 BEEP
200 CURSOR 18,12:PRINT CHR$(48+X)
210 BEEP
220 IF (A=B) AND (A=X) THEN 290
230 N=-M
240 CURSOR 0,21:PRINT "MISE ";M:PRINT
"RESULTAT ";N
250 FOR L=1 TO 500
260 NEXT L
270 P=P+N
280 GOTO 20
290 N=M*50
300 IF A<>7 THEN 320
310 N=M*500
320 GOTO 240
330 X=INT(9*RND(1)+1)
340 RETURN
350 CLS
360 PRINT"VOUS AVEZ ";P:PRINT"A BIENTO
T"
```

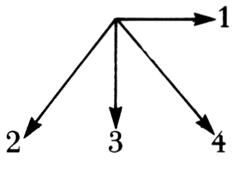
Le jeu de la fève

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : réflexion (deux joueurs).
Difficulté du jeu : moyenne.

LE JEU :
Présentant une certaine analogie avec le jeu de WYTTHOF, cette version offre en fait 4 possibilités de déplacement ; la position de départ est ici fixée en (Ø, Ø), et pourrait éventuellement être modifiée.
Les directions de déplacement disponibles sont : droite, bas, diagonale bas et droite, diagonale bas et gauche.
On joue en principe avec la règle : le premier arrivé en (9, 9) a perdu.

LE PROGRAMME :
Les boucles emboîtées J (3Ø-9Ø) et I (4Ø-8Ø) dessinent le terrain de jeu.
La ligne 14Ø réalise le test de fin de partie.
La ligne 2ØØ permet d'exécuter l'un des sous-programmes de déplacement.
Les lignes 21Ø et 22Ø assurent le changement de joueur.

POUR JOUER :
On lance le programme par RUN, et le terrain de jeu s'affiche, ainsi que la question DIR 1, 2, 3, 4 ? ; les 4 directions proposées correspondent aux déplacements suivants :



Le joueur 1 entre son déplacement.
Le mouvement correspondant est réalisé, puis on obtient l'affichage JR 2, et DIR...
Le joueur 2 procède de même, et ainsi de suite, jusqu'à ce que l'un des joueurs arrive en (9, 9) auquel cas le jeu se termine.

EXTENSIONS POSSIBLES :
On peut imaginer d'introduire en début de partie des cases pièges (chacun essayant alors de faire venir l'autre sur l'une de ces cases).
On peut aussi jouer avec plusieurs symboles, si possible de couleurs différentes, déplacés simultanément, envisager des terrains de jeu triangulaires, aux dimensions arbitraires, etc.

LE JEU DE LA FEVE

```
10 CLS
20 A=0
30 FOR J=0 TO 9
40 FOR I=0 TO 9
50 CURSOR 2*I+6,5:PRINT CHR$(48+I)
60 CURSOR 4,J+7:PRINT CHR$(48+J)
70 CURSOR 2*I+6,J+7:PRINT CHR$(144)
80 NEXT I
90 NEXT J
100 X=7
110 Y=6
120 J=1
130 CURSOR X,Y:PRINT "*"
140 IF (X>=16) AND (Y>=24) THEN 240
150 T$="JOUEUR NO. "+STR$(J)+" DIR 1,2
,3,4 ?"
160 CURSOR 3,22:PRINT T$
170 BEEP
180 K$=INKEY$
190 IF K$="" THEN 180
200 ON ASC(K$)-48 GOSUB 280,270,250,30
0
210 J=J+1
220 IF J=3 THEN 120
230 GOTO 130
240 END
250 Y=Y+2
260 RETURN
270 GOSUB 250
280 X=X+1
290 RETURN
300 X=X-1
310 Y=Y+2
320 RETURN
```

Jeu de l'oie aléatoire

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : hasard (deux joueurs).
Difficulté du jeu : facile.

LE JEU :
Le jeu reprend la très ancienne idée du jeu de l'oie, mais en y ajoutant la possibilité offerte par l'ordinateur de choisir au hasard les cases "à problème".
La règle du jeu est sans grand intérêt, puisque les joueurs n'interviennent à aucun moment dans le déroulement de la partie, qui est entièrement gérée par le YENO ; il suffira d'appuyer sur une touche après chaque lancer de dé, et de regarder ce qui se passe.

LE PROGRAMME :
La ligne 5Ø permet d'appeler le sous-programme (29Ø-43Ø) qui réalise l'affichage du terrain de jeu, le contrôle de la position des joueurs et l'affichage de cette position. Cet affichage sera d'ailleurs refait en cours de partie, pour tenir compte du déplacement réellement effectué (qui n'est pas nécessairement celui indiqué par le dé).
La ligne 6Ø s'occupe du changement de joueur, et la ligne 7Ø effectue le lancer de dé.
La ligne 9Ø traite le cas où l'un des joueurs termine.
La ligne 15Ø tire un nombre aléatoire entre Ø et 3, qui servira s'il y a lieu pour déterminer le vrai déplacement du joueur, en envoyant dans l'un des sous-programmes de "chance".
Les lignes 29Ø à 35Ø font avancer les pions, et les lignes 36Ø à 43Ø redessinent à chaque affichage le terrain de jeu en entier.

POUR JOUER :
On lancera le jeu avec RUN. Le terrain de jeu est alors dessiné, puis on voit apparaître l'affichage JOUEUR 1 :...
On devra alors appuyer sur une touche quelconque pour que le coup du joueur numéro 1 soit pris en compte ; ne pas s'inquiéter du temps de réaction, qui pourra sembler un peu long.
Le déplacement indiqué par les dés sera exécuté, puis au bout de quelque temps, un second déplacement sera éventuellement effectué : c'est le déplacement "CHANCE". On obtiendra alors l'affichage JOUEUR 2 :...
Le joueur 2 appuie à son tour sur une touche quelconque et ainsi de suite...
Le joueur gagnant sera signalé par un affichage BRAVO en fin de partie.
On peut aussi s'arrêter à tout moment en tapant BREAK.

EXTENSIONS POSSIBLES :
On peut agrandir le terrain de jeu, introduire davantage d'aléas, changer leur nature. On pourrait aussi choisir les cases "chance" en début de jeu, ajouter des affichages en conséquence, etc.

JEU DE L'OIE ALEATOIRE

```
10 X=1
20 Y=X-1
30 J=1
40 I=X
50 GOSUB 290
60 J=ABS(J-1)
70 I=INT(6*RND(1)+1)
80 CURSOR 1,20
90 IF (X<>32) AND (Y<>32) THEN 110
100 PRINT"BRAVO ";2-J
110 PRINT"JR ";J+1;":";I
120 S$=INKEY$
130 IF S$="" THEN 120
140 GOSUB 290
150 I=INT(4*RND(1))
160 IF RND(1)>=.5 THEN 50
170 ON I+1 GOSUB 190,200,220,240
180 GOTO 50
190 RETURN
200 I=-I
210 RETURN
220 I=I*2
230 RETURN
240 IF J<>0 THEN 270
250 X=INT(2+31*RND(1))
260 GOTO 280
270 Y=INT(2+31*RND(1))
280 RETURN
290 IF J>0 THEN 330
300 X=X+1*(X-(I<=32)):X=ABS(X)
310 IF X<=32 THEN 330
320 X=32
330 Y=Y+J*I*(Y-(I<=32)):Y=ABS(Y)
340 IF Y<=32 THEN 360
350 Y=32
360 CLS
370 CURSOR 2,13:FOR I=2 TO 6:PRINT CHR
$(51+I);:NEXT I
380 CURSOR 7,13:FOR I=7 TO 32:PRINT CH
R$(58+I);:NEXT I
390 CURSOR 1,14:FOR I=1 TO 32:PRINT CH
R$(229);:NEXT I
400 CURSOR 1,15:FOR I=1 TO 32:PRINT CH
R$(229);:NEXT I
410 CURSOR X,14:PRINT"1"
420 CURSOR Y,15:PRINT"2"
430 RETURN
```

Niveau 2

Nom du programme	*Thèmes et particularités*
Un dé	Tirage d'un dé, avec graphisme.
Visions	Un jeu sur les mots, pour l'ordinateur !
Plus grand, plus petit	Un jeu de déduction intéressant, pour un joueur.
L'électricien fou	Logique et difficile, pour un seul joueur.
La course au 2Ø	Jeu de réflexion, pour 2 joueurs ou plus.
Dessin	Une méthode de codage originale.
Dé truqué	A utiliser, si vous êtes malhonnête, dans d'autres programmes.
Le jeu de saute-mouton	Un jeu de réflexion, pour 2 joueurs.
Lettres	Un jeu de mots, assez simple (2 joueurs).
Espion	Offrez un YENO à votre fiancée, et dites-lui votre amour en code...
Accrochez les wagons	Pour les plus jeunes ; très visuel.
Tirage d'un dé truqué	Révisez vos connaissances en statistiques...
Ordre	Un peu d'astuce pour remettre en ordre, le plus vite possible...
Course de chevaux	Assez visuel ; un jeu de hasard pour plusieurs joueurs.
Le jeu de Marienbad	Un jeu de logique pour 2 joueurs, dans une version étendue.
Idem	Un jeu de calcul mental, pour 2 joueurs (plusieurs niveaux).
Mastermind	Avec des chiffres, mais très classique ; un seul joueur.
A travers	Un jeu de stratégie élémentaire, pour 2 joueurs.
Reines sur l'échiquier	Un vieux problème, pour vous aider à réfléchir.
Le jeu de la vie	Un grand classique : deux versions, manuelle et automatique.

Dimensionnement d'un tableau Recherche d'un élément dans un tableau et dans une chaîne

L'INSTRUCTION DIM, OU INSTRUCTION DE DIMENSIONNEMENT D'UN TABLEAU

Les structures de tableaux sont très couramment utilisées en programmation, se révélant même indispensables dans un grand nombre de cas. Si vous voulez, par exemple, entrer vos recettes et vos dépenses pour tous les jours du mois (on supposera pour simplifier des mois de trente jours) vous pourriez définir 3Ø variables appelées PREMIER JOUR, DEUXIEME JOUR, etc. Il est plus facile, et plus rationnel d'utiliser une boucle I, I variant de 1 à 3Ø, et une variable R (I) pour les recettes, et D (I) pour les dépenses ; ce faisant, vous avez, sans le savoir, utilisé une structure de tableau à une dimension ; ceci ne pourra être admis par le YENO que si vous l'avez auparavant prévenu que vous alliez utiliser un tel tableau, en donnant une instruction DIM R (3Ø) pour les recettes, et DIM D (3Ø) pour les dépenses.

On pourrait aussi imaginer de conserver dépenses et recettes dans un même tableau, en lui donnant cette fois deux dimensions : par exemple DIM RD (3Ø, 2) (ou DIM RD (2, 3Ø)). Ainsi, avec DIM RD (3Ø, 2) on pourra convenir que RD (I, 1) représente les recettes, et RD (I, 2) les dépenses. Des tableaux de plus de 2 dimensions sont également possibles.

On peut aussi concevoir des tableaux dont les éléments sont des chaînes de caractères, et non plus des nombres.

Ainsi DIM NOM$ (5) réserve 5 emplacements en mémoire, pour des mots.

LA RECHERCHE D'UN ELEMENT DANS UN TABLEAU : R (I)

Si l'on a un tableau à une dimension, défini par une instruction comme DIM R (3Ø), le Ième élément (numérique) de ce tableau est repéré par R (I) ; il sera alors manipulé exactement comme une variable numérique ordinaire (affectation, calcul, tests, etc.).

Ainsi INPUT R (3) permet d'entrer à la main la valeur du 3e élément numérique du tableau R. PRINT R (5) permet d'afficher le 5e élément du même tableau R ; etc.

EXTRACTION D'UN ELEMENT DANS UNE CHAINE

- RIGHT$ (R$,P) extrait de la chaîne R$ les P caractères de droite.
- LEFT$ (R$,P) extrait de la chaîne R$ les P caractères de gauche.
- MID$ (R$,P,L) extrait de la chaîne R$, la sous-chaîne de L caractères à partir du Pième inclus.
- LEN (R$) donne le nombre de caractères de la chaîne R$.

Exemples :
A$ = MID$ ("EXEMPLE", 3, 4) affecte à la variable A$ la valeur "EMPL".
A$ = LEFT$ ("EXEMPLE", 2) affecte à A$ la valeur "EX".
X = LEN ("EXEMPLE") affecte à X la valeur 7.

Un dé

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : tirage d'un dé (avec graphisme).
Difficulté du jeu : aucune.

LE JEU :
Le programme propose simplement le lancer d'un dé aléatoire, et sa représentation graphique en grand format.
Il pourra facilement être utilisé en sous-programme, pour tout type de jeu faisant appel au lancer d'un ou de plusieurs dés.

LE PROGRAMME :
La ligne 2Ø utilise le générateur aléatoire du YENO pour tirer un entier de 1 à 6.
Les lignes 4Ø à 7Ø dessinent les bords du dé, et les lignes 3Ø et 8Ø dessinent le haut et le bas.
Les 7 lignes 1ØØ, 12Ø, 14Ø, 15Ø, 17Ø, 19Ø et 21Ø sont utilisées pour le dessin des points.
La ligne 23Ø permet le cas échéant de refaire un tirage.

POUR JOUER :
On lance le dé par RUN, le résultat est affiché sous forme symbolique, comme sur un véritable dé.
La question AUTRE TIRAGE (O ou N) est ensuite affichée.
Si l'on répond O, un autre lancer sera effectué.

EXTENSIONS POSSIBLES :
On pourra utiliser ce programme en tant que sous-programme chaque fois que l'on a besoin de simuler un lancer de dé.

UN DE

```
10 CLS
20 A=INT(1+6*RND(1))
30 CURSOR 8,5:FOR I=1 TO 9:PRINT CHR$(
229);:NEXT I
40 FOR I=1 TO 7
50 CURSOR 8,5+I:PRINT CHR$(229)
60 CURSOR 16,5+I:PRINT CHR$(229)
70 NEXT I
80 CURSOR 8,13:FOR I=1 TO 9:PRINT CHR$
(229);:NEXT I
90 IF A=1 THEN 110
100 CURSOR 10,7:PRINT CHR$(249)
110 IF A<4 THEN 130
120 CURSOR 14,7:PRINT CHR$(249)
130 IF A<>6 THEN 160
140 CURSOR 10,9:PRINT CHR$(249)
150 CURSOR 14,9:PRINT CHR$(249)
160 IF A/2=INT(A/2) THEN 180
170 CURSOR 12,9:PRINT CHR$(249)
180 IF A<4 THEN 200
190 CURSOR 10,11:PRINT CHR$(249)
200 IF A=1 THEN 220
210 CURSOR 14,11:PRINT CHR$(249)
220 CURSOR 0,20:INPUT "AUTRE TIRAGE (O
/N) ";R$
230 IF R$="O" THEN 10
```

Visions

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : dessin.
Difficulté du jeu : facile.

LE JEU :
Le YENO vous permet, pour la première fois, d'entrer dans son intimité en vous laissant entrevoir sa propre représentation du monde ! Vous allez, en effet, lui proposer des mots, n'importe lesquels, et il va vous répondre en vous proposant une représentation graphique de ce que représente ce mot pour lui. Le jeu sera beaucoup plus amusant si vous le pratiquez à plusieurs, et vous pourrez quelquefois avoir des surprises.

LE PROGRAMME :
Une boucle I (3Ø-12Ø) se charge de réaliser les dessins.
La ligne 4Ø permet de transformer en nombre chaque lettre du mot entré au début (ligne 2Ø). Ce nombre sera ensuite traduit en base 2 par une boucle J (lignes 5Ø à 11Ø) ; selon le résultat de cette transformation, un point sera imprimé, ou non (ligne 8Ø).

POUR JOUER :
On débute l'exécution du programme par RUN, ce qui amène l'affichage de la question QUEL DESSIN. On répond en entrant un mot quelconque qui sera transformé en dessin par le YENO dès que l'on aura frappé CR. Pour recommencer, il suffira de frapper une touche quelconque et BREAK. Je vous laisse savourer le dessin qu'a réalisé le YENO lorsqu'il a connu le prénom de l'auteur de ce livre.

EXTENSIONS POSSIBLES :
La même idée peut être reprise en introduisant des éléments de symétrie dans les dessins, en ajoutant la couleur. On peut aussi imaginer de coder par cette méthode un certain nombre de dessins, que l'on introduirait au fur et à mesure dans la machine et l'on pourra alors créer un jeu dans lequel le YENO deviendra de plus en plus savant. On peut aussi réaliser un jeu à deux joueurs dans lequel le YENO vous demande de trouver le mot qui correspond à un certain dessin, etc.

VISIONS

```
10 CLS
20 INPUT "QUEL DESSIN ";R$
30 FOR I=1 TO LEN(R$)
40 N=9*(ASC(MID$(R$,I,1))-27)-340
50 J=1
60 S=(N/2)-INT(N/2)
70 N=INT(N/2)
80 IF S=0 THEN 100
90 CURSOR J+12,I+5:PRINT CHR$(229)
100 J=J+1
110 IF J-9<>0 THEN 60
120 NEXT I
130 S$=INKEY$
140 IF S$="" THEN 130
150 GOTO 10
```

Plus grand, plus petit

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu de déduction (un seul joueur).
Difficulté du jeu : difficile.

LE JEU :
Il s'inspire d'un principe légèrement différent de celui du Mastermind ; il s'agit également de deviner un nombre (de 1 à 7 chiffres), mais les indications dont on dispose sont cette fois d'une part le nombre de chiffres qui, dans le nombre essayé, sont supérieurs à ceux du nombre à deviner, et, d'autre part, le nombre de chiffres qui sont inférieurs. Si, par exemple, le nombre à deviner est 3Ø54, et que l'on essaie 4251, la machine répondra 2 plus grands (ce sont le 4 et le 2), et 1 plus petit (c'est le 1).

LE PROGRAMME :
La ligne 3Ø interdit de demander des nombres de plus de 8 chiffres.
Le nombre à deviner est placé en ligne 5Ø, dans une variable numérique X.
Les comparaisons avec le nombre essayé sont faites à la ligne 8Ø (qui teste si l'on a trouvé juste), et dans la boucle I (11Ø-14Ø) où l'on fait la comparaison "lettre à lettre".
L'affichage du résultat est réalisé grâce à la ligne 15Ø.
Le compteur du nombre d'essais est en ligne 16Ø.

POUR JOUER :
Le lancement du programme par RUN provoque l'affichage NBRE DE CHIFFRES.
On entre un nombre inférieur à 8 et on appuie sur la touche CR.
Le numéro de l'essai s'affiche, ainsi que la question NBRE PRESUME ; on répond en entrant un nombre, suivi de CR.
Le résultat est alors affiché sous la forme ... CHIFFRES PLUS GRANDS et ... CHIFFRES PLUS PETITS.
S'il n'est pas exact, on recommence (attention à entrer uniquement des nombres ayant autant de chiffres que le nombre à deviner).
Si l'on a trouvé, le YENO affichera TROUVE... EN... ESSAIS. Puis AUTRE PARTIE (O ou N) ? ; on fera O suivi de CR pour commencer une autre partie.

EXTENSIONS POSSIBLES :
On pourra, comme pour le Mastermind, imaginer un jeu analogue avec des lettres, des couleurs, ou même en utilisant d'autres symbolismes.
On peut aussi prévoir de laisser jouer le YENO qui tiendra alors le rôle de décodeur.

PLUS GRAND, PLUS PETIT

```
10 CLS
20 INPUT "NBRE DE CHIFFRES ";C
30 IF C>8 THEN 10
40 E=1
50 X=INT(9*RND(1)*10^(C-1))
60 PRINT:PRINT:PRINT"ESSAI N0 ";E:PRIN
T:PRINT:PRINT:PRINT
70 INPUT "NBRE PRESUME ";P
80 IF P=X THEN 180
90 A=0
100 B=A
110 FOR I=2 TO C+1
120 A=A-(MID$(STR$(P),I,1)<MID$(STR$(X
),I,1))
130 B=B-(MID$(STR$(P),I,1)>MID$(STR$(X
),I,1))
140 NEXT I
150 PRINT:PRINT:PRINT:PRINT:PRINTA;" C
HIFFRES PLUS GRANDS":PRINT:PRINTB;" CH
IFFRES PLUS PETITS"
160 E=E+1
170 GOTO 60
180 CLS
190 PRINT "TROUVE ";X;" EN ";E;" ESSAI
S":PRINT:PRINT
200 INPUT "UNE AUTRE PARTIE (O/N) ";R$

210 IF R$="O" THEN 10
```

L'électricien fou

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : solitaire.
Difficulté du jeu : assez difficile.

LE JEU :
Un immeuble comportant 1Ø fenêtres, numérotées de Ø à 9, apparaît sur l'écran. Le joueur devra à chaque essai manœuvrer l'un des interrupteurs, en indiquant son numéro (de Ø à 9).
En principe, la fenêtre correspondante devrait s'éteindre, mais il est aussi possible qu'une autre s'allume (ou s'éteigne). Il s'agit, bien sûr, d'essayer d'obtenir l'extinction des 1Ø fenêtres, avec le moins d'essais possible.

LE PROGRAMME :
Une table A contiendra l'état (allumé ou éteint) des 1Ø fenêtres.
La boucle I (4Ø-1ØØ) et l'autre boucle I (13Ø-19Ø) dessinent l'immeuble dans son état initial.
La ligne 2ØØ regarde si tout est éteint, auquel cas l'on ira en ligne 46Ø afficher le nombre d'essais.
La ligne 23Ø attend l'indication de l'interrupteur manœuvré au moyen d'un INKEY$.
Les lignes 28Ø à 4ØØ définissent le résultat de la manœuvre.

POUR JOUER :
RUN provoque l'affichage de l'immeuble, avec toutes les fenêtres allumées.
A la question INTER, on répond en indiquant le numéro de l'interrupteur à manœuvrer.
Le résultat de la manœuvre est alors affiché sur l'écran (des fenêtres s'allument ou s'éteignent).
On continue ainsi jusqu'à ce que toutes les fenêtres soient éteintes, et l'on obtient alors l'affichage du nombre d'essais réalisés.

EXTENSIONS POSSIBLES :
On pourra envisager un graphisme très amélioré avec si possible l'utilisation de la couleur. On pourra aussi définir un jeu à deux joueurs, augmenter le nombre de fenêtres, modifier la formule définissant l'allumage ou l'extinction...

L'ELECTRICIEN FOU

```
10 CLS
20 S=1
30 M=0
40 FOR I=S TO 23
50 CURSOR I+5,6:PRINT CHR$(229)
60 CURSOR I+5,6+S:PRINT CHR$(229)
70 CURSOR I+5,6+2*S:PRINT CHR$(229)
80 CURSOR I+5,9:PRINT CHR$(229)
90 CURSOR I+5,10:PRINT CHR$(229)
100 NEXT I
110 CURSOR 4,18:PRINT"INTERRUPTEUR N0
:"
120 T=0
130 FOR I=S TO 10
140 T=T+A(I)
150 CURSOR 2*I+S+5,2*S+6:PRINT CHR$(22
9)
160 IF A(I)<>0 THEN 180
170 CURSOR 2*I+S+5,6+2*S:PRINT CHR$(32
)
180 CURSOR 2*I+S+5,10:PRINT CHR$(47+I)

190 NEXT I
200 IF T=10 THEN 410
210 CURSOR 27,18:PRINT CHR$(32)
220 BEEP
230 K$=INKEY$
240 IF K$="" THEN 230
250 BEEP
260 CURSOR 27,18:PRINT K$
270 N=ASC(K$)-48
280 N=N+S
290 M=M+S
300 IF A(N)=S THEN 390
310 A(N)=S
320 R=TAN(RND(1)+N/RND(1)-N)-SIN(RND(1
)/N)+336*SIN(8*N)
330 N=INT(10*(R-INT(R)))
340 IF N<>0 THEN 360
350 N=INT(S+10*RND(1))
360 IF A(N)=S THEN 390
370 A(N)=S
380 GOTO 120
390 A(N)=0
400 GOTO 320
410 CURSOR 1,20:PRINT "TROUVE EN ";M;"
 ESSAIS"
```

La course au 2Ø

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : logique (deux joueurs, ou plus).
Difficulté du jeu : moyenne.

LE JEU :
Il est possible d'imaginer pour ce jeu diverses règles, et nous n'avons retenu ici que la règle usuelle pour deux joueurs, qui en fait un jeu apparenté aux jeux de déplacement, ou d'allumettes : 4 tas de 2Ø vont apparaître en début de partie, et chaque joueur peut prendre autant de carrés qu'il le désire, mais ceci dans un seul tas à la fois. Celui qui prend le dernier carré est considéré comme perdant.

LE PROGRAMME :
La boucle I (3Ø-7Ø) dessine 4 rangées de 2Ø carrés.
La variable F, dimensionnée en ligne 2Ø, servira à enregistrer le nombre de carrés pris dans chaque rangée.
La variable T contiendra le nombre total de carrés enlevés.
La ligne 5Ø imprime les numéros de ligne.
La ligne 12Ø va contrôler qu'il reste suffisamment de carrés sur la ligne indiquée.
La boucle I (16Ø-22Ø) redessine le terrain de jeu après modification.
La ligne 26Ø teste la fin de partie.

POUR JOUER :
Lancer le programme par RUN ; le terrain de jeu s'affiche, puis les inscriptions JR 1 et LGN, puis CASES ? On répond en entrant un numéro de ligne (entre 1 et 4), et le nombre de carrés à enlever suivi de CR. Les carrés sont alors supprimés, et c'est au tour du deuxième joueur.
La partie se termine, lorsque le dernier carré a été enlevé, par l'affichage du joueur gagnant.

EXTENSIONS POSSIBLES :
Outre une amélioration de la qualité du graphique et des contrôles, de nombreuses variantes sont envisageables, en modifiant si besoin est, les règles de départ : on peut autoriser l'enlèvement d'un nombre impair de carrés, ou alterner pair et impair, ou tirer au hasard le nombre à enlever, ou encore jouer à plusieurs...

LA COURSE AU 20

```
10 CLS
20 DIM F(4)
30 FOR I=1 TO 4
40 F(I)=0
50 CURSOR 6,9+2*I:PRINT CHR$(48+I)
60 CURSOR 9,9+2*I:FOR J=1 TO 20:PRINT
CHR$(229);:NEXT J
70 NEXT I
80 T=0
90 B=1
100 CURSOR 0,19:PRINT "JOUEUR ";B:PRIN
T
110 INPUT "LIGNE, CASES ?";L,C
120 IF (C+F(L)>20) OR (L>4) THEN 110
130 F(L)=F(L)+C
140 T=T+C
150 CLS
160 FOR I=1 TO 4
170 CURSOR 6,9+2*I:PRINT CHR$(48+I)
180 IF F(I)=20 THEN 200
190 CURSOR 9+F(I),9+2*I:FOR J=1 TO 20-
F(I):PRINT CHR$(229);:NEXT J
200 IF F(I)=0 THEN 220
210 CURSOR 9,9+2*I:FOR J=1 TO F(I):PRI
NT"*";:NEXT J
220 NEXT I
230 B=B+1
240 IF B<>3 THEN 260
250 B=1
260 IF T<80 THEN 100
270 CLS
280 PRINT "LE JOUEUR ";B;" A GAGNE"
```

Dessin

REFERENCES :
Mode SLOW ou FAST.
Difficulté de programmation : facile.
Catégorie de jeu : dessin.
Difficulté du jeu : moyenne.

LE JEU :
C'est surtout une méthode un peu originale de codage d'un dessin, qui pourra être avantageusement utilisée dans certains programmes.

LE PROGRAMME :
Le codage du dessin est, dans le programme retenu, figé dans une variable A$, définie en ligne 5Ø à l'aide des caractères graphiques A$, B$ et C$ (lignes 2Ø à 5Ø).
On pourra éventuellement essayer d'autres dessins en entrant directement leurs codes : il suffit de modifier la ligne concernée par INPUT A$, B$ ...
La boucle N (6Ø-14Ø) réalise le décodage de X$.
La ligne 1ØØ détermine les positions de la tête d'écriture.
La ligne 12Ø détecte le caractère Ø, qui sert à indiquer le retour à la ligne.

POUR JOUER :
Le dessin devra auparavant être défini par une variable X$, en utilisant les caractères redéfinis, et en donnant la position d'impression de chaque caractère, par rapport au début de la ligne, avec un chiffre de 1 à 9, ou une lettre (A : 1Ø ; B : 11 ; C : 12 ; etc.).
Le passage à la ligne suivante sera indiqué par le caractère Ø.

EXTENSIONS POSSIBLES :
La fonction PATTERN permettra de définir un graphique plus fin, on pourra aussi envisager des indications pour la couleur, imaginer aussi un jeu dans lequel chaque joueur devrait essayer de coder un dessin donné le plus rapidement possible, etc.

Remarque : Pour entrer directement dans une chaîne un caractère redéfini au moyen de PATTERN, on peut se servir de la touche CTRL.
Par exemple : CTRL A correspond au caractère de code ASCII 129. Se référer au manuel d'utilisation pour de plus amples détails.

DESSIN

```
10 CLS
20 A$=CHR$(229)
30 B$=CHR$(155)
40 C$=CHR$(236)
50 X$="A":FOR I=1 TO 18:X$=X$+A$:NEXT
I:X$=X$+"0A"+A$+"K"+B$+"R"+A$+"0A"+A$+
"K"+B$+"R"+A$+"0A"+A$+"K"+B$+"R"+A$+"0
3":FOR I=1 TO 18:X$=X$+A$:NEXT I:X$=X$
+B$+B$+B$+B$+B$+A$+"09"+C$+"B"+C$+"M"+
C$+"O"+C$+"0A"+C$+"N"+C$+"0"
60 FOR N=1 TO LEN(X$)
70 S=ASC(MID$(X$,N,1))
80 IF S<128 THEN 100
90 PRINT MID$(X$,N,1);
100 IF ((S<49) OR (S>57)) AND ((S<65)
OR (S>90)) THEN 120
110 PRINT TAB(S-48+7*(S>=65));
120 IF S<>48 THEN 140
130 PRINT""
140 NEXT N
150 CURSOR 0,22
```

Dé truqué

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : facile.

LE JEU :
On veut essayer de fabriquer un dé truqué selon sa propre convenance. La méthode proposée ici consiste à définir au départ une probabilité de réalisation pour chaque type de tirage (soit la probabilité d'obtenir un 1, un 2, un 3, etc. que chacune de ces probabilités soit comprise entre Ø et 1, et que la somme de toutes les probabilités soit un nombre très voisin de 1).

LE PROGRAMME :
Il utilise le générateur aléatoire du YENO supposé de qualité convenable (on pourra aisément utiliser tout autre générateur, si besoin était). La boucle I (4Ø-1ØØ) s'occupe de l'entrée de chacune des probabilités, avec vérification à chaque entrée et sur le total (ligne 9Ø) ; chaque probabilité est conservée dans une variable indicée E (I) : ligne 6Ø du programme.
Le lancer de dé truqué est effectué à l'aide d'une autre boucle I (19Ø-21Ø), la ligne permettant le trucage se trouvant en 2ØØ. Un nouveau tirage pourra être effectué sur simple pression d'une touche (lignes 23Ø et 24Ø).

POUR JOUER :
Après le lancement de l'exécution par RUN, on obtient l'affichage de la question PROBABILITE DU DE 1, à laquelle on devra répondre par un nombre compris entre Ø et 1. Il conviendra de ne pas oublier que le total de toutes les probabilités entrées ne devra en aucun cas dépasser 1.1 (tolérance de Ø.1).
Des messages d'erreurs pourront être affichés et le programme sera éventuellement relancé. Lorsque toutes les probabilités auront été définies, le premier tirage sera automatiquement effectué, et il suffira de presser une touche quelconque (sauf BREAK) pour effectuer les tirages suivants. On arrête le programme en tapant BREAK.

EXTENSIONS POSSIBLES :
Il sera bien évidemment très tentant d'utiliser sournoisement ce petit programme pour améliorer vos chances de gain dans un jeu de hasard utilisant des lancers de dés. On pourra alors envisager de figer les données définissant le dé truqué dans une variable A$, etc. On pourra aussi envisager de présenter les résultats correspondants à une série de lancers de dés sous une forme spectaculaire, qui mette le trucage en évidence.

DE TRUQUE

```
10 CLS
20 P=0
30 DIM E(6)
40 FOR I=1 TO 6
50 PRINT"PROBABILITE DU DE ";I;
60 INPUT E(I)
70 PRINT
80 P=P+E(I)
90 IF (E(I)<0) OR (E(I)>1) OR (P>1.1)
THEN 260
100 NEXT I
110 PRINT
120 FOR I=1 TO 6
130 PRINT "DE ";I,E(I)
140 NEXT I
150 IF P<.9 THEN 10
160 FOR I=1 TO 300
170 NEXT I
180 N=INT(RND(1)*6+1)
190 FOR I=1 TO 6
200 IF (N=I) AND (RND(1)>E(I)) THEN 18
0
210 NEXT I
220 PRINT:PRINT"TIRAGE ";N
230 K$=INKEY$
240 IF K$="" THEN 230
250 GOTO 180
260 PRINT:PRINT:PRINT"IMPOSSIBLE :0<=
P <=1"
270 FOR L=1 TO 700
280 NEXT L
290 P=P-E(I)
300 CLS
310 GOTO 50
```

Le jeu de saute-mouton

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : facile.

LE JEU :
Bien que l'idée en soit très simple, il ne sera pas forcément évident de découvrir une stratégie convenable. On dispose de 3 moutons “>” et de 3 moutons “<”, qui occupent 8 cases ; les moutons peuvent se déplacer d'une case à une case libre. Plusieurs règles pourront intervenir, selon votre propre estimation de l'intérêt de telle ou telle : soit l'on contraint les moutons à se placer sur la première case disponible, en sautant éventuellement par-dessus les moutons adverses, soit on leur laisse choisir leur case d'arrivée, ce qui peut rendre les parties interminables, à vous de choisir.

LE PROGRAMME :
La boucle I (6Ø-9Ø) initialise une table A (I), qui contient les positions des moutons. La boucle I (11Ø-17Ø) dessine le terrain de jeu et les moutons.
Les lignes 18Ø et 19Ø s'occupent du texte et des entrées.
La ligne 2ØØ contrôle la validité du coup. Le changement de joueur est réalisé en ligne 23Ø.

POUR JOUER :
On commence l'exécution du programme par RUN. On obtient alors l'affichage JR 1, qui indique le numéro du joueur, suivi de la question DEBUT, FIN ?
On répond en indiquant le numéro du pion à déplacer, puis le numéro de la case d'arrivée, suivi de CR.
Le déplacement est alors réalisé ; puis le deuxième joueur procède de la même façon. Le gagnant est en principe celui qui réussit à placer ses trois moutons côte à côte sur les emplacements initialement occupés par l'adversaire.

EXTENSIONS POSSIBLES :
On commencera par rajouter un contrôle du joueur gagnant.
On peut également améliorer la présentation en dessinant de vrais moutons, rendre le jeu plus attractif en les numérotant, et en imposant un certain ordre à l'arrivée.
Enfin, on pourra définir une stratégie pour l'ordinateur, en figeant les règles.

LE JEU DE SAUTE-MOUTON

```
10 CLS
20 B=1
30 DIM A(8)
40 A(1)=0
50 A(8)=0
60 FOR I=B TO 3
70 A(B+I)=B
80 A(4+I)=-B
90 NEXT I
100 H=B
110 FOR I=B TO 8
120 CURSOR 3*I+B+3,17:PRINT CHR$(48+I)

130 IF A(I)<>B THEN 150
140 CURSOR 3*I+B+3,15:PRINT">"
150 IF A(I)<>-B THEN 170
160 CURSOR 3*I+B+3,15:PRINT"<"
170 NEXT I
180 CURSOR 0,20:PRINT"JOUEUR ";(H=B)+2

190 INPUT "DEBUT,FIN ?";D,F
200 IF (A(D)<>H)OR (A(F)<>1-B) THEN 18
0
210 A(D)=1-B
220 A(F)=B*H
230 H=-H
240 CLS
250 GOTO 110
```

Lettres

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : moyenne.

LE JEU :
La règle est fortement inspirée d'un jeu de lettres très connu en France, pratiqué notamment dans une émission télévisée (les chiffres et les lettres). Le YENO choisit au hasard un certain nombre de symboles (voyelles et consonnes) dans l'alphabet latin, et chacun des joueurs devra indiquer la longueur du mot qu'il a réussi à obtenir (l'appareil ne vérifiera pas si le mot est correct !).
Le score total de chaque joueur est affiché ensuite.

LE PROGRAMME :
La boucle I (7Ø-11Ø) choisit dans l'alphabet entre 7 et 12 lettres, en fonction de la variable aléatoire L, tirée en ligne 4Ø. Pour éviter d'avoir trop de consonnes, on utilise en ligne 1ØØ une petite astuce, qui consiste à refaire au hasard un tirage dans la table A$, contenant des voyelles (on peut en avoir plus ou moins en utilisant d'autres RND).
La boucle 12Ø-19Ø demande à chacun des joueurs la longueur de son mot. Les lignes 2ØØ et 21Ø comparent les résultats et totalisent le score de chaque joueur, affiché par la ligne 22Ø.

POUR JOUER :
On lance l'exécution du jeu avec RUN.
Les lettres sont alors affichées, puis la question JR 1 - LONG à laquelle le premier joueur répond en donnant la longueur du mot qu'il a obtenu (l'autre joueur devra toutefois vérifier si le mot est correct) ; puis le joueur 2 fera de même et le score de chaque joueur sera alors affiché.
Pour continuer la partie, il suffira d'appuyer sur une touche quelconque (sauf la touche BREAK qui permettra de finir).

EXTENSIONS POSSIBLES :
On pourra bien entendu songer à introduire une temporisation avant de demander leurs réponses aux joueurs.
On pourra aussi entrer non point la longueur du mot, mais le mot lui-même, l'adversaire devant alors répondre à la question CORRECT OU NON ? On peut aussi étendre le jeu à plus de deux joueurs. Et pourquoi ne pas essayer un jeu contre le YENO, auquel on aurait auparavant entré une liste de mots assez longue, dont il afficherait les lettres dans le désordre, avec éventuellement quelques lettres de complément choisies au hasard.

LETTRES

```
10 S=0
20 J=S
30 CLS
40 L=INT(6*RND(1)+7)
50 A$="AEIOUE"
60 PRINT"LETTRES :":PRINT:PRINT:PRINT:
PRINT:PRINT:PRINT:PRINT:PRINT
70 FOR I=1 TO L
80 CURSOR 7+2*I,10:PRINT CHR$(INT(26*R
ND(1)+65))
90 IF RND(1)>=.4 THEN 110
100 CURSOR 7+2*I,10:PRINT MID$(A$,INT(
6*RND(1)+1),1)
110 NEXT I
120 I=1
130 T$="JOUEUR"+STR$(I)+":LONGUEUR "
140 CURSOR 0,19:PRINT SPC(30)
150 CURSOR 0,19:PRINT T$;:INPUT L
160 IF I-1<>0 THEN 180
170 P=L
180 I=I+1
190 IF I<3 THEN 130
200 S=S-(P*(P>L))
210 J=J-(L*(L>P))
220 PRINT:PRINT:PRINT:PRINT"JOUEUR 1:"
;S,"JOUEUR 2:";J;
230 K$=INKEY$:IF K$="" THEN 230
240 GOTO 30
```

Espion

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu de réflexion.
Difficulté du jeu : facile.

LE JEU :
Le programme peut servir simultanément de codeur et de décodeur de messages secrets, suivant une méthode très facile à mettre en œuvre sur ce type d'appareil.
Un petit générateur aléatoire situé en fin de programme est utilisé pour modifier de façon arbitraire chaque lettre du message ; on devra bien surveiller à utiliser toujours le même nombre (noyau) pour commencer le codage ou le décodage de deux messages identiques.

LE PROGRAMME :
Le noyau initial (qui sera un entier quelconque) est entré en ligne 2Ø ; le message est entré à la ligne 4Ø.
La ligne 5Ø permet de choisir entre les options de codage et de décodage.
La boucle I (6Ø-15Ø) est utilisée soit pour effectuer le codage, soit pour effectuer le décodage.
La ligne 8Ø appelle le générateur aléatoire.
La ligne 9Ø réalise la transposition des lettres.
Les lignes 1ØØ et 12Ø vérifient que l'on reste bien dans les limites autorisées.
Les résultats seront imprimés par la ligne 16Ø.
Les lignes 18Ø à 21Ø contiennent le générateur.

POUR JOUER :
Lancer le programme par RUN.
Entrer ensuite un entier N qui servira de noyau pour le générateur (il faudra le retenir pour le décodage).
Puis taper le message, suivi de CR.
Choisir l'option C pour CODER, et l'option D pour DECODER, et taper CR.
Le message codé ou décodé, selon le cas, sera alors affiché.

EXTENSIONS POSSIBLES :
On pourra étendre le codage à d'autres signes que les lettres de l'alphabet, utiliser éventuellement plusieurs générateurs, introduire dans le message un codage du ou des nombres ayant servi de générateurs, etc.

ESPION

```
10 CLS
20 INPUT "CODE ";N
30 A=1/N
40 INPUT"MESSAGE: ";A$:Z=LEN(A$)
50 INPUT "C)ODEUR OU D)ECODEUR ";R$
60 FOR I=1 TO LEN(A$)
70 T=ASC(MID$(A$,I,1))
80 GOSUB 180
90 T=T+A*((R$="D")-(R$="C"))
100 IF T<=90 THEN 120
110 T=T-26
120 IF T>=65 THEN 140
130 T=T+26
140 A$=MID$(A$,1,I-1)+CHR$(T)+MID$(A$,
I+1,Z-1)
150 NEXT I
160 PRINT"RESULTAT: ";A$
170 END
180 A=(A+3.14159)*8
190 A=A-INT(A)
200 A=INT(26*A+1)
210 RETURN
```

Accrochez les wagons

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : solitaire.
Difficulté du jeu : facile.

LE JEU :
Il faut essayer d'accrocher, le plus rapidement possible, la locomotive à son wagon, qui apparaissent tous deux en début de partie, en deux endroits plus ou moins éloignés de l'écran. On devra pour cela déplacer la locomotive en indiquant la suite des mouvements à effectuer (sous forme codée : H pour une case vers le haut ; B pour une case vers le bas ; D pour une case vers la droite et G pour une case vers la gauche).
On tapera, par exemple : BBGDHHH.

LE PROGRAMME :
Les lignes 1Ø à 4Ø définissent les caractères qui seront utilisés pour dessiner la locomotive et le wagon.
La position initiale de la locomotive est définie par les lignes 5Ø et 6Ø, celle du wagon par les lignes 7Ø et 8Ø.
La ligne 9Ø évite qu'ils soient initialement trop proches l'un de l'autre.
Les lignes 11Ø à 16Ø réalisent l'affichage du wagon et les lignes 17Ø à 24Ø celui de la locomotive.
Le test de fin est en ligne 25Ø.
Les mouvements à effectuer sont entrés dans une variable alphanumérique R$, en ligne 26Ø, et sont décodés dans une boucle I (3ØØ-33Ø) pour être exécutés en lignes 34Ø et 35Ø.

POUR JOUER :
RUN permet d'afficher les positions de départ de la locomotive et du wagon.
On tape alors la série des déplacements à effectuer, sous la forme DDDHHHBHGG, suivie de CR.
La locomotive sera alors affichée dans sa nouvelle position, et l'on recommence, jusqu'à ce qu'elle soit accrochée au wagon.
On obtiendra alors l'affichage du score.

EXTENSIONS POSSIBLES :
Il sera intéressant de prévoir la perte de la partie si, par accident, au cours du déplacement, la locomotive vient en collision avec le wagon.
On pourra aussi prévoir plusieurs wagons, et un ordre d'accrochage, jouer à plusieurs avec plusieurs locomotives, dessiner les voies ferrées, introduire des obstacles à contourner obligatoirement d'un certain côté, etc.

ACCROCHEZ LES WAGONS

```
10 PATTERN C# 131,"000000000F0F0F0F"
20 PATTERN C# 130,"FFFFFFFFFFFFFFFF"
30 PATTERN C# 129, "F0F0F0F0FFFFFFFF"
40 PATTERN C# 132,"00000000FFFFFFFF"
50 LX=INT(21*RND(1)+2)
60 LY=INT(25*RND(1)+3)
70 WX=INT(21*RND(1)+2)
80 WY=INT(25*RND(1)+3)
90 IF ABS(LX-WX)<4 THEN 50
100 CLS
110 CURSOR WY,WX:PRINT CHR$(129)
120 CURSOR WY+1,WX:PRINT CHR$(132)
130 CURSOR WY+2,WX:PRINT CHR$(129)
140 CURSOR WY,WX+1:PRINT CHR$(48)
150 CURSOR WY+1,WX+1:PRINT CHR$(32)
160 CURSOR WY+2,WX+1:PRINT CHR$(48)
170 CURSOR LY,LX-1:PRINT CHR$(131)
180 CURSOR LY+1,LX-1:PRINT CHR$(132);C
HR$(132)
190 CURSOR LY+3,LX-1:PRINT CHR$(131)
200 CURSOR LY,LX:PRINT CHR$(132)
210 CURSOR LY+1,LX:PRINT CHR$(130);CHR
$(130);CHR$(130)
220 CURSOR LY+1,LX+1:PRINT CHR$(48)
230 CURSOR LY+2,LX+1:PRINT CHR$(32)
240 CURSOR LY+3,LX+1:PRINT CHR$(48)
250 IF (LX=WX) AND (WY=LY-3) THEN 370
260 CURSOR 0,22:INPUT "CHEMIN A SUIVRE
 :";R$
270 X=0
280 Y=X
290 N=N+1
300 FOR I=1 TO LEN(R$)
310 X=X+(MID$(R$,I,1)="B")*(LX<20)-(MI
D$(R$,I,1)="H")*(LX>2)
320 Y=Y+(MID$(R$,I,1)="D")*(LY<28)-(MI
D$(R$,I,1)="G")*(LY>2)
330 NEXT I
340 LX=LX+X
350 LY=LY+Y
360 GOTO 100
370 CURSOR 0,21:PRINT "BRAVO, TROUVE E
N ";N;" COUPS"
```

Tirage d'un dé truqué

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion.
Difficulté du jeu : moyenne.

LE JEU :
Il reprend une idée d'un programme précédent (DE TRUQUE) dans une application statistique : on fabrique un dé volontairement truqué, que l'on va lancer un certain nombre de fois, et l'on dresse un tableau des résultats.

LE PROGRAMME :
Les probabilités désirées pour chaque tirage sont introduites par une boucle I (4Ø-9Ø)avec toutes les vérifications de validité nécessaires. Le nombre de tirages demandés est introduit en ligne 12Ø.
La boucle J (14Ø-22Ø) comptabilise les tirages.
La boucle I (16Ø-18Ø) assure la conformité de chaque tirage avec les probabilités introduites en début de jeu.
Les résultats sont comptabilisés en ligne 19Ø, et affichés grâce aux lignes 2ØØ et 21Ø.

POUR JOUER :
Commencer l'exécution en frappant RUN, puis à la question PROBABILITE DU DE 1, introduire un nombre compris entre Ø et 1.
Faire de même pour les dés 2, 3, etc., sans oublier que le total des probabilités introduites devra être très proche de 1.
Ensuite, à l'affichage de la question NOMBRE DE TIRAGES, on introduira un entier suffisamment grand, suivi de CR. L'affichage des résultats mettra d'autant plus de temps à apparaître que l'entier introduit sera grand.
Les résultats sont affichés sous les numéros des tirages. Ainsi 32.3 apparaissant sous le 1 signifie que le tirage de 1, auquel on avait affecté une probabilité de Ø.3, a été réalisé 32 fois.

EXTENSIONS POSSIBLES :
On peut envisager d'améliorer la représentation des résultats en utilisant par exemple un histogramme, ou toute autre méthode graphiquement plus parlante.

TIRAGE D'UN DE TRUQUE

```
10 DIM E(6)
20 CLS
30 P=0
40 FOR I=1 TO 6
50 PRINT "PROBABILITE DU DE "+STR$(I)+
":";
60 INPUT E(I)
70 P=P+E(I)
80 IF (E(I)<0) OR (E(I)>1) OR (P>1.1)
THEN 240
90 NEXT I
100 IF P<.9 THEN 20
110 PRINT :PRINT :PRINT :PRINT
120 INPUT "NOMBRE DE TIRAGE ";P
130 FOR K=1 TO 9:PRINT :NEXT K
140 FOR J=1 TO P
150 N=INT(6*RND(1)+1)
160 FOR I=1 TO 6
170 IF (N=I) AND (RND(1)>(E(I)-INT(E(I
)))) THEN 160
180 NEXT I
190 E(N)=E(N)+1
200 CURSOR 5*N-1,22:PRINT E(N)
210 CURSOR 5*N+1,20:PRINT CHR$(48+N)
220 NEXT J
230 END
240 PRINT :PRINT :PRINT "IMPOSSIBLE :
0<=P<=1"
250 FOR K=1 TO 250
260 NEXT K
270 GOTO 60
```

Ordre

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de réflexion.
Difficulté du jeu : moyenne.

LE JEU :
Les 1Ø premières lettres de l'alphabet sont affichées dans le désordre ; il faut réussir à les remettre dans l'ordre, avec le moins de coups possible, par des permutations successives de deux des lettres ; mais ceci a aussi pour inconvénient de faire permuter autour d'elles les deux lettres qui entourent celles-ci, suivant le mécanisme expliqué ci-dessous :

```
B A D G H J I C F E
Ø 1 2 3 4 5 6 7 8 9   : permutation 38

B A D F H J I E G C
Ø 1 2 3 4 5 6 7 8 9   : résultat
```

LE PROGRAMME :
Les boucles emboîtées I (3Ø-9Ø) et J (5Ø-7Ø) définissent une variable alphanumérique A$, contenant les 1Ø premières lettres de l'alphabet dans le désordre.
La ligne 11Ø affiche A$, et les numéros de lettres à utiliser pour la permutation.
On entre la permutation en ligne 12Ø.
La ligne 14Ø effectue la première inversion.
La ligne 16Ø effectue la deuxième.
La ligne 17Ø comptabilise les résultats.
La ligne 18Ø contrôle les résultats.

POUR JOUER :
Lancer le programme par RUN.
Le mot initial est affiché, chaque lettre étant repérée par un numéro de Ø à 9.
A la question : INVERSION (I et J), on répond en entrant deux nombres x, y ; x est le numéro de la première lettre, et y le numéro de la seconde, sachant que x < y. Les deux inversions sont alors successivement effectuées, et l'on recommence.
Le nombre de coups sera affiché en fin de partie.

EXTENSIONS POSSIBLES :
On peut imaginer un jeu contre l'ordinateur, qui évaluerait le score du joueur en fonction de ses propres performances ; on peut aussi modifier les règles qui régissent les permutations.

ORDRE

```
10 CLS
20 A$=""
30 FOR I=1 TO 10
40 R$=CHR$(INT(10*RND(1)+65))
50 FOR J=1 TO I-1
60 IF R$=MID$(A$,J,1) THEN 40
70 NEXT J
80 A$=A$+R$
90 NEXT I
100 S=1
110 PRINT :PRINT :PRINT :PRINT ,A$:PRI
NT ,"0123456789":PRINT "INVERSION(I,J)
";" ESSAI ";S
120 INPUT I,J
130 IF J<I THEN 120
140 A$=MID$(A$,1,I)+MID$(A$,J+1,1)+MID
$(A$,I+2,J-I-1)+MID$(A$,I+1,1)+MID$(A$
,J+2,9-J)
150 IF J=9 THEN 170
160 A$=MID$(A$,1,J-1)+MID$(A$,J+2,1)+M
ID$(A$,J+1,1)+MID$(A$,J,1)+MID$(A$,J+3
,8-J)
170 S=S+1
180 IF A$<>"ABCDEFGHIJ" THEN 110
190 PRINT :PRINT :PRINT "TROUVE EN";S-
1;"COUPS"
```

Course de chevaux

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : hasard (plusieurs joueurs).
Difficulté du jeu : aucune.

LE JEU :
On entrera un nombre de chevaux compris entre 1 et 1Ø, et la course débutera immédiatement.
Il pourra y avoir des ex-aequo, car l'on tiendra compte du fait que le YENO s'occupe de faire avancer les chevaux les uns après les autres, en commençant par le numéro 1 ; le classement sera affiché au fur et à mesure des arrivées (et en tenant compte des ex-aequo).

LE PROGRAMME :
Une première boucle I (7Ø-21Ø) affiche les chevaux dans leur position de départ.
Une deuxième boucle I (1ØØ-19Ø) les fera avancer ; la position de chaque cheval sera repérée par un chiffre ou une lettre dans une variable alphanumérique A$.
La ligne 12Ø fera avancer les chevaux.
La ligne 13Ø gère les arrivées.
Les lignes 15Ø et 16Ø réalisent les impressions.
Le sous-programme 28Ø-32Ø fait avancer le compteur des arrivées, et imprime les résultats.

POUR JOUER :
On lance le programme par RUN.
A la question NBRE DE CHEVAUX ? on répond par un entier de 1 à 1Ø, suivi de CR.
La course se déroulera ensuite automatiquement.

EXTENSIONS POSSIBLES :
On pourra dessiner les chevaux avec un graphisme plus soigné, et les différencier par leur couleur.
On pourra aussi imaginer tout un système de paris, qui interviendra avant le départ de chaque course.

COURSE DE CHEVAUX

```
10 CLS
20 A$=""
30 INPUT " NBRE DE CHEVAUX :";N
40 P=0
50 T=1
60 A=P
70 FOR I=T TO N
80 A$=A$+"1"
90 GOTO 150
100 FOR I=1 TO N
110 IF MID$(A$,I,1)="K" THEN 210
120 A$=LEFT$(A$,I-1)+CHR$(ASC(MID$(A$,
I,1))+INT(4*RND(1)+1))+RIGHT$(A$,LEN(A
$)-I)
130 IF MID$(A$,I,1)<"K" THEN 150
140 GOSUB 280
150 CURSOR ASC(MID$(A$,I,1))-49,I+I:PR
INT "    "
160 CURSOR ASC(MID$(A$,I,1))-45,I+I:PR
INT CHR$(96+I)
170 FOR IP=I TO200
180 NEXT IP
190 NEXT I
200 GOTO 220
210 NEXT I
220 P=P+T
230 T=0
240 FOR IP=1 TO 400
250 NEXT IP
260 IF A<N THEN 100
270 GOTO 330
280 A$=LEFT$(A$,I-1)+"K"+RIGHT$(A$,LEN
(A$)-I)
290 CURSOR 3,I+I:PRINT CHR$(48+P)
300 T=1
310 A=A+T
320 RETURN
330 END
```

Le jeu de Marienbad

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : logique (2 joueurs).
Difficulté du jeu : moyenne.

LE JEU :
Un jeu rendu très célèbre grâce à un film qui le popularisa, présenté toutefois ici dans une version plus conforme à ses origines, à savoir le jeu de NIM.
En effet, il est possible, en début de partie, de choisir le nombre de tas que l'on veut faire et le nombre d'allumettes que l'on veut mettre dans chaque tas.
Chaque joueur joue à son tour, et celui qui retire la dernière allumette a perdu.

LE PROGRAMME :
La ligne 6Ø dimensionne une variable A (I) qui contiendra le nombre d'allumettes de chaque tas.
La boucle I (7Ø-11Ø) permet d'entrer ce nombre.
Les boucles I (14Ø-16Ø, 18Ø-2ØØ) impriment les tas.
La ligne 22Ø assure le changement de joueur.
La ligne 25Ø contrôle si le coup est autorisé.
La ligne 32Ø contrôle la fin de la partie.

POUR JOUER :
On lance le programme avec RUN, puis on entre le nombre de tas (entre 1 et 8), suivi de CR.
On entrera ensuite le nombre d'allumettes dans chaque tas (faire à chaque entrée N suivi de CR). On obtiendra alors l'affichage des différents tas, avec le numéro du joueur concerné (1 ou 2), ainsi que le message N° DU TAS ; on répond en tapant le numéro du tas dans lequel on désire faire un retrait suivi de CR. On obtient ensuite la question NBRE A OTER, et l'on répond en indiquant le nombre d'allumettes que l'on désire enlever du tas concerné.
Puis ce sera au tour du deuxième joueur, et l'on jouera alternativement jusqu'à ce qu'il n'y ait plus d'allumettes disponibles.
En fin de partie, on obtiendra l'affichage du numéro du joueur gagnant.

EXTENSIONS POSSIBLES :
On peut désigner chaque joueur par son nom, faire un graphisme plus élaboré, écrire un programme permettant au YENO de servir de partenaire, etc.

LE JEU DE MARIENBAD

```
10 CLS
20 INPUT  " NBRE DE TAS :";N
30 PRINT :PRINT
40 S=0
50 IF N>=8 THEN 10
60 DIM A(7)
70 FOR I=1 TO N
80 PRINT "TAS"+STR$(I)+",NBRE D'ALLUME
TTE";
90 INPUT A(I)
100 S=S+A(I)
110 NEXT I
120 H=1
130 CLS
140 FOR I=1 TO N
150 PRINT TAB(4*I+1);A(I);
160 NEXT I
170 PRINT :PRINT "TAS";
180 FOR I=1 TO N
190 PRINT TAB(1+4*I);I;
200 NEXT I
210 PRINT :PRINT :PRINT :PRINT TAB(4);
"JOUEUR ";H:PRINT
220 IF H=1 THEN 240
230 H=0
240 INPUT "PRISE DANS LE TAS ";T
250 IF (T>N) OR (A(T)=0) THEN 240
260 PRINT :PRINT
270 INPUT "NBRE A OTER:";O
280 IF (O>A(T)) OR (O=0) THEN 270
290 A(T)=A(T)-O
300 S=S-O
310 H=H+1
320 IF S=0 THEN 340
330 GOTO 130
340 PRINT :PRINT :PRINT "LE JOUEUR ";H
;"A GAGNE"
```

Idem

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de calcul mental (2 joueurs).
Difficulté du jeu : selon le niveau choisi.

LE JEU :
Le YENO choisit un nombre de n chiffres (selon le niveau de difficulté indiqué au début). Ce nombre (le même au début pour les 2 joueurs) sera régulièrement décrémenté à chaque tour, d'une quantité également choisie au hasard par YENO en début de jeu ; vous devez, à chaque coup, choisir une colonne (les dizaines, les centaines, etc.) et ajouter dans cette colonne un certain entier ; le but est d'arriver à obtenir à un moment l'affichage d'un nombre ayant tous ses chiffres identiques.
Chaque résultat sera affiché un court instant, après quoi la main passera au joueur suivant, qui repartira de son précédent résultat.

LE PROGRAMME :
Une variable numérique N est dimensionnée à 2 en ligne 1Ø, pour contenir les résultats successifs des deux joueurs.
La ligne 4Ø permettra d'entrer le niveau de difficulté choisi.
Le nombre initial est confectionné par la ligne 7Ø. La quantité qui interviendra dans les décrémentations est obtenue en ligne 1ØØ.
La ligne 12Ø teste si le résultat n'est pas trop petit.
La ligne 14Ø affiche le numéro du joueur et le nombre. Le jeu du joueur est entré en ligne 15Ø (deux chiffres).
La ligne 16Ø incrémente la colonne choisie, et la ligne 17Ø décrémente le nombre de la quantité choisie au départ.
Le résultat sera affiché un court instant (lignes 18Ø-2ØØ). La boucle I (22Ø-24Ø) permet de vérifier si tous les chiffres sont identiques.
L'impression et la fin de partie sont effectuées en ligne 25Ø. Les lignes 28Ø-3ØØ s'occupent du changement de joueur.

POUR JOUER :
Lancer l'exécution du programme par RUN.
Le nombre initial sera affiché pour le joueur n° 1, qui joue en tapant x, y suivi de CR ; x est le chiffre à ajouter, et y est le numéro de la colonne où il faut l'ajouter.
Le résultat (provisoire s'il est trop petit) sera alors affiché un court instant, et le trait passe au deuxième joueur, et ainsi de suite. On peut arrêter à tout moment en tapant BREAK.

EXTENSIONS POSSIBLES :
On pourra étendre le jeu à plus de 2 joueurs, améliorer le choix du nombre intervenant dans la décrémentation, améliorer les divers contrôles, laisser les différents résultats affichés, etc.

IDEM

```
10 DIM M(2)
20 H=1
30 CLS
40 INPUT "NIVEAU (1-7) :";N
50 U=10
60 P=0
70 M(1)=INT((U^N-U^(N-1))*RND(1)+U^(N-
1))
80 M(2)=M(1)
90 A=INT(2*U^(N-1)*RND(1)+U^(N-1))
100 CLS
110 P=P-(H=1)
120 IF M(H)>=U^(N-1) THEN 140
130 M(H)=INT(U^N-ABS(M(H)))
140 PRINT "JOUEUR ";H,"ESSAI ";P:PRINT
 :PRINT :PRINT "NOMBRE";TAB(15-N-INT(L
GT(M(H))));M(H):PRINT TAB(8);"7654321
":PRINT :PRINT :PRINT
150 INPUT "VOTRE JEU (N,C) ?  ";R1,R2
160 M(H)=M(H)+R1*U^(R2-1)
170 M(H)=INT(M(H)-A)
180 PRINT :PRINT :PRINT "RESULTAT ";M(
H)
190 FOR I=1 TO 400
200 NEXT I
210 R$=STR$(M(H))
220 FOR I=2 TO LEN(R$)
230 IF MID$(R$,2,1)<>MID$(R$,I,1) THEN
 270
240 NEXT I
250 PRINT :PRINT :PRINT "BRAVO, JOUEUR
 ";H;":";P;" ESSAIS "
260 END
270 H=H+1
280 IF H<3 THEN 100
290 H=1
300 GOTO 100
```

Mastermind

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : logique (un seul joueur).
Difficulté du jeu : difficile.

LE JEU :
Devenu un grand classique des jeux sur micro-ordinateur, ce jeu développe le raisonnement déductif ; il faut réussir à trouver le plus vite possible un nombre de n chiffres (on pourra prendre n compris entre 1 et 5), en disposant à chaque essai de renseignements.
En effet, le YENO répondra en indiquant d'une part le nombre de chiffres qui sont, dans le nombre proposé, exactement à la même place que dans le nombre à découvrir, et, d'autre part, le nombre de chiffres qui figurent dans le nombre à découvrir sans être exactement au même emplacement.

LE PROGRAMME :
Le choix du nombre de chiffres du nombre à découvrir est effectué aux lignes 2Ø et 3Ø.
Le nombre est fabriqué à la ligne 5Ø (il pourra y avoir des chiffres répétés).
L'affichage est préparé par la ligne 6Ø.
La saisie de EC se fait au moyen de INPUT (ligne 11Ø).
La ligne 8Ø efface la ligne précédente.
La ligne 14Ø s'occupe de la fin de partie.
Une première boucle I (17Ø-24Ø) teste les chiffres qui sont à la bonne place.
Une deuxième boucle J (19Ø-21Ø) teste les chiffres qui sont présents et mal placés.
Les compteurs sont aux lignes 2ØØ et 23Ø.
La ligne 25Ø affiche le résultat, et la ligne 26Ø comptabilise les essais.

POUR JOUER :
On tape RUN et l'on choisit la longueur du nombre désiré (de 1 à 6), suivie de CR.
Après l'affichage NO. NOMBRE BON PLACE, on introduit son propre essai, suivi de CR.
On obtient immédiatement la réponse (en cas de double chiffre, les résultats peuvent être un peu délicats à interpréter).
On introduit ensuite le nouvel essai, et ainsi de suite.
Lorsque l'on a trouvé, on obtiendra l'affichage TROUVE EN ... COUPS.
On peut sortir du programme à tout moment en tapant BREAK.

EXTENSIONS POSSIBLES :
On peut chercher à éviter la présence de chiffres doubles dans le nombre à trouver, on peut aussi prévoir un changement de page automatique, avec rappel des résultats précédents ; on pourra aussi réaliser des jeux analogues avec des mots, ou d'autres symboles (couleur, taille...).

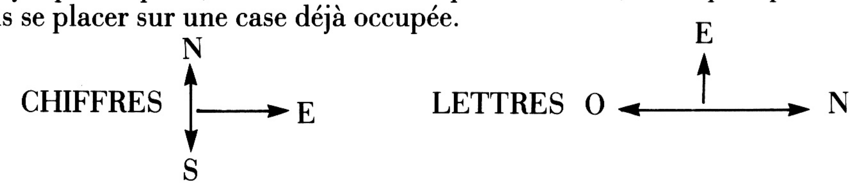
MASTERMIND

```
10 CLS
20 INPUT "NOMBRE DE CHIFFRE ";N
30 IF N>=6 THEN 10
40 C=1
50 E=INT(RND(1)*9^N+10^(N-1))
60 PRINT:PRINT :PRINT:PRINT "NO.      N
OMBRE      BON        MAL PLACE":PRINT :
PRINT
70 E$=""
80 CURSOR 8,20:PRINT "
                               "
90 CURSOR 0,20
100 BEEP
110 INPUT "        >";EC
120 X=0
130 Y=X
140 IF E<>EC THEN 170
150 CURSOR 0,20:PRINT"TROUVE EN ";C;"
COUPS"
160 GOTO 280
170 FOR I=2 TO N+1
180 IF MID$(STR$(EC),I,1)=MID$(STR$(E
),I,1) THEN 230
190 FOR J=2 TO N+1
200 IF (J<>I) AND (MID$(STR$(EC),I,1))
=(MID$(STR$(E),J,1)) THEN Y=Y+1
210 NEXT J
220 GOTO 240
230 X=X+1
240 NEXT I
250 CURSOR 0,5+C:PRINTC;TAB(9);EC;TAB(
21);X;TAB(31);Y
260 C=C+1
270 GOTO 70
280 END
```

A travers

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : moyenne.

LE JEU :
Sur l'écran apparaît un carré de 9 cases, avec sur le bord vertical des lettres et sur le bord horizontal des chiffres.
L'un des joueurs manipule les chiffres, l'autre les lettres. 3 déplacements sont possibles pour les chiffres, et 3 autres pour les lettres. Les lettres peuvent se déplacer vers le haut, à droite et à gauche et les chiffres vers la droite, en haut ou en bas.
Le gagnant est le premier qui arrive à faire sortir tous ses symboles (les lettres sortent en haut, et les chiffres à droite).
Il n'y a pas de prise, et les sauts ne sont pas autorisés ; on ne peut pas non plus se placer sur une case déjà occupée.



LE PROGRAMME :
Les positions des chiffres et des lettres sont enregistrées dans une table N (4, 4) ; les lignes 1Ø à 11Ø définissent les valeurs de départ.
Les lignes 13Ø à 2ØØ dessinent le terrain de jeu et la ligne 21Ø affiche le joueur qui a la main.
La ligne 23Ø contrôle le coup joué, de même que les boucles I (24Ø-28Ø) et J (25Ø-27Ø).
Le jeu proprement dit est géré par les lignes 3ØØ à 37Ø et le changement de joueur est assuré par la ligne 38Ø.

POUR JOUER :
Le terrain de jeu sera affiché après la frappe de RUN et l'ordinateur attend le coup du joueur désigné par les pions qu'il possède.
On entrera d'abord le chiffre ou la lettre représentant le pion à déplacer, suivi d'une indication de direction sous la forme N (pour Nord), S (pour Sud), E (pour Est) ou O (pour Ouest).
On pourra ainsi faire 2E suivi de CR.
Si le coup est invalide, il ne sera pas exécuté, et le code du joueur restera à l'affichage ; s'il est autorisé, le déplacement sera effectué, et l'on obtient l'affichage correspondant au deuxième joueur.
Le premier qui aura réussi à faire disparaître ses 3 pièces aura gagné.

EXTENSIONS POSSIBLES :
On peut envisager d'étendre le jeu avec davantage de chiffres et de lettres et d'améliorer les contrôles. On peut aussi définir une stratégie pour le YENO.

```
              A TRAVERS

10 K=1
20 J$(0)="LETTRES"
30 J$(1)="CHIFFRES"
40 F=2
50 S=4
60 DIM N(4,4)
70 FOR I=K TO S
80 N(I,K)=48+I
90 N(S,I)=63+I
100 NEXT I
110 N(S,K)=0
120 CLS
130 FOR I=1 TO S
140 FOR J=1 TO S
150 IF N(J,I)<>0 THEN 180
160 CURSOR F*I+12,F*J+8:PRINT "."
170 GOTO 190
180 CURSOR F*I+12,F*J+8:PRINT CHR$(N(J
,I))
190 NEXTJ
200 NEXT I
210 CURSOR 0,20:PRINT "JR. AYANT LES "
+J$(K)+" ";
220 INPUT R$
230 IF (K<>0 AND (MID$(R$,F,1)="O")) O
R (K=0 AND (MID$(R$,F,1)="S")) THEN 21
0
240 FOR I=1 TO S
250 FOR J=1 TO S
260 IF N(I,J)=ASC(MID$(R$,1,1)) THEN 3
00
270 NEXT J
280 NEXT I
290 GOTO 210
300 A=I
310 B=J
320 I=I-(MID$(R$,F,1)="S")+(MID$(R$,F,
1)="N")
330 J=J-(MID$(R$,F,1)="E")+(MID$(R$,F,
1)="O")
340 IF (I=0) AND (K=0) OR(J>S) AND(K<>
0) THEN 370
350 IF N(I,J)<>0 THEN 210
360 N(I,J)=ASC(MID$(R$,1,1))
370 N(A,B)=0
380 K=ABS(K-1)
390 GOTO120
```

Reines sur l'échiquier

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion.
Difficulté du jeu : difficile.

LE JEU :
Le programme utilise un algorithme très performant pour résoudre un problème qui a donné du fil à retordre à des chercheurs, même aussi célèbres que le mathématicien GAUSS, qui n'avait pas pu découvrir toutes les solutions. Il s'agit de trouver toutes les manières de disposer des reines sur un échiquier, de telle sorte qu'aucune ne puisse en prendre une autre.
Le programme retenu est très souple, puisque vous disposez même de la possibilité de choisir la taille de votre échiquier, et les résultats seront affichés à la fois sous forme graphique et sous forme numérique.

LE PROGRAMME :
La ligne 2Ø vous permet de choisir la taille de l'échiquier.
Une variable D est surdimensionnée en ligne 3Ø.
Une boucle d'exploration principale I (4Ø-1ØØ) contient une sous-boucle J (7Ø-9Ø) qui réalise le contrôle de non-alignement (ligne 8Ø).
Si, au cours de l'exploration, on rencontre un alignement, on revient en arrière, en utilisant les lignes 23Ø à 26Ø et on recommence.
La boucle F (12Ø-2ØØ) assure l'impression de l'échiquier, et la boucle T (15Ø-17Ø), que l'on pourra éventuellement supprimer, assure l'impression du résultat sous forme d'un code numérique correspondant à la position des pièces dans chaque colonne.
La ligne 22Ø permet un arrêt après l'affichage.

POUR JOUER :
On lance le programme par RUN, puis on tape la dimension de l'échiquier désirée, et on fait CR.
Il suffit d'attendre, avec patience (plus l'échiquier sera grand, plus l'attente sera longue).
Il faudra taper BREAK en fin d'exécution.

EXTENSIONS POSSIBLES :
On peut envisager de trouver des algorithmes pour tous les problèmes analogues (position de tours, déplacement d'un cavalier, etc.).

REINES SUR L'ECHIQUIER

```
10 CLS
20 INPUT " DIMENSION ? ";S
30 DIM D(20)
40 FOR I=1 TO S
50 D(I)=1
60 IF I=1 THEN 100
70 FOR J=1 TO I-1
80 IF (D(I)=D(J)) OR (ABS(D(I)-D(J))=I
-J) THEN 230
90 NEXT J
100 NEXT I
110 CLS
120 FOR F =1 TO S
130 CURSOR 2*F+3,3:PRINT CHR$(48+F)
140 CURSOR 3,2*F+3:PRINT CHR$(48+F)
150 FOR T=1 TO S
160 CURSOR 2*T+3,2*F+3:PRINT" "
170 NEXT T
180 CURSOR 2*D(F)+3,2*F+3:PRINT CHR$(2
29)
190 CURSOR 3+F,20:PRINT CHR$(48+D(F))
200 NEXT F
210 BEEP
220 GOTO 210
230 D(I)=D(I)+1
240 IF D(I)<=S THEN 60
250 I=I-1
260 IF I<>0 THEN 230
```

Le jeu de la vie

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu solitaire (2 versions).
Difficulté du jeu : facile.

LE JEU :
La version présentée ici respecte les règles posées à l'origine par le mathématicien J.H. CONWAY.
Une certaine population est définie au début du jeu soit par le YENO (version entièrement automatique) soit par vous-même (version manuelle). Cette population va évoluer sous vos yeux, en respectant certaines règles, très élémentaires, de naissance, de vie et de mort. L'évolution se poursuivra jusqu'à ce que l'on décide d'arrêter le programme.

LE PROGRAMME :
Deux tables (B et C) sont utilisées pour enregistrer l'état de la population. Les boucles F (6Ø-2ØØ) et G (7Ø-19Ø) sont utilisées en début de jeu pour définir la population initiale. Le caclul de la nouvelle population est ensuite réalisé à l'aide d'une quadruple boucle : 6Ø-2ØØ ; 7Ø-19Ø ; 9Ø-14Ø ; 1ØØ-13Ø. La ligne 12Ø réalise la gestion du nouvel état suivant les règles. L'affichage de ce nouvel état est obtenu en réutilisant la double boucle d'affichage F (22Ø-31Ø) et G (23Ø-3ØØ).
La ligne 32Ø permet de recommencer.

POUR JOUER :
Dans la version automatique, il suffit de lancer le programme par RUN et de regarder. Pour arrêter, il faudra taper BREAK. Dans la version manuelle, la configuration de départ sera entrée, après le lancement du programme par RUN. L'évolution se déroulera ensuite de façon automatique.

EXTENSIONS POSSIBLES :
Il sera facile d'augmenter les dimensions des tables B et C. On pourra aussi introduire la couleur, améliorer la précision du dessin en redéfinissant des caractères.

LE JEU DE LA VIE (AUTOMATIQUE)

```
10 CLS
20 INPUT "DIMENSION (max=10)  ? ";A
30 D=1
40 I=0
50 GOTO 210
60 FOR F=D TO A
70 FOR G=D TO A
80 S=0
90 FOR I=-D TO D
100 FOR J=-D TO D
110 IF (F+J=0) AND (G+I=0) AND (F+J=A+
D) AND(G+I=A+D) THEN 130
120 S=S+B(F+J,D+I)
130 NEXT J
140 NEXT I
150 S=2*S-B(F,G)
160 C(F,G)=0
170 IF (S<5)AND (S>7) THEN 190
180 C(F,G)=D
190 NEXT G
200 NEXT F
210 CLS
220 FOR F=D TO A
230 FOR G=D TO A
240 IF I<>0 THEN 270
250 IF RND(1)> 5 THEN 270
255 BEEP
260 C(F,G)=D
270 IF C(F,G)<>1 THEN 290
280 CURSOR 2*G+2,2*F:PRINT CHR$(229)
290 B(F,G)=C(F,G)
300 NEXT G
310 NEXT F
320 GOTO 60
```

LE JEU DE LA VIE (MANUEL)

```
10 CLS
20 INPUT "DIMENSION (max=10)  ? ";A
30 D=1
40 I=0
50 GOTO 210
60 FOR F=D TO A
70 FOR G=D TO A
80 S=0
90 FOR I=-D TO D
100 FOR J=-D TO D
110 IF (F+J=0) AND (G+I=0) AND (F+J=A+
D) AND(G+I=A+D) THEN 130
120 S=S+B(F+J,D+I)
130 NEXT J
140 NEXT I
150 S=2*S-B(F,G)
160 C(F,G)=0
170 IF (S<5) OR (S>7) THEN 190
180 C(F,G)=D
190 NEXT G
200 NEXT F
210 CLS
220 FOR F=D TO A
230 FOR G=D TO A
240 IF I<>0 THEN 270
250 CURSOR 2,2*F:PRINT ">"
260 CURSOR 2*G+2,1:PRINT "V"
270 FOR P=1 TO 300
280 NEXT P
290 C(F,G)=1
300 K$=INKEY$
310 IF K$="" THEN C(F,G)=0
320 CURSOR 2,2*F:PRINT " "
330 CURSOR 2*G+2,1:PRINT " "
340 IF C(F,G)<>1 THEN 360
350 CURSOR 2*G+2,2*F:PRINT CHR$(229)
360 B(F,G)=C(F,G)
370 NEXT G
380 NEXT F
390 GOTO 60
```

Niveau 3

Nom du programme	*Thèmes et particularités*
Losanges	Pour vous aider à vous endormir ...
Echiquier	Des bases graphiques, pour vos propres créations.
Couloirs	Un jeu de logique, à découvrir.
Jour de la semaine	Profitez-en pour faire le point sur les anniversaires.
Le pendu	Un jeu de lettres très classique (2 joueurs).
Bridge	Mettez au point vos propres problèmes de bridge.
Horloge 1	Ne jetez pas votre vieille télé : transformez-la en horloge !
Le loup et les agneaux	Un jeu de réflexion, pour 2 joueurs.
Belote	Vous n'aimez pas le bridge ? Alors, utilisez Belote !
Un taquin de calcul	Un bon entraînement au calcul (2 joueurs).
Poker	Une base de départ, pour mettre au point votre programme.
Hanoï	Un conseil : commencez petit ! ! !
Musique	Révisez vos notions de solfège, grâce au YENO.
Pendu inversé	Cette fois, c'est YENO qui essaie de deviner votre mot !
Mathieu	Vous ne savez pas dessiner ? Laisser faire YENO.
Machinamot	Peut-être YENO trouvera-t-il des mots qui n'existent pas ?
Compose	Pour voir, pour apprendre, et pour entendre ...
Nicomaque	Entraînement au calcul mental (1 ou plusieurs joueurs).
Permutations	Un jeu de réflexion, pas toujours évident.
Tour de cartes	Et, pour finir, un peu de prestidigitation, par YENO.
Parcours du cavalier	Très spectaculaire et instructif à plus d'un titre.
Mur de briques	Un grand classique, dans une version très complète.

Manipulation des chaînes et graphiques haute résolution

TRAITEMENT DES CHAINES DE CARACTERES : MID$

Nous avons déjà parlé, au niveau 2, de l'instruction MID$ en la comparant aux instructions équivalentes dans d'autres Basic.

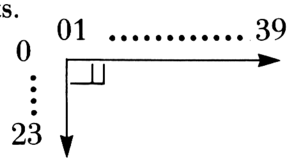
Rappelons que MID$ (EX$, P, L) extrait de la chaîne EX$ la sous-chaîne débutant au Pième élément et de la longueur L. Si P + L est supérieur à la longueur de EX$, la sous-chaîne est tronquée ; elle est vide si P est supérieur à cette même longueur. Si la valeur de P ou de L est négative ou nulle, le programme stoppe en affichant un message d'erreur.

En Basic YENO, le signe + permet la concaténation de 2 chaînes.
Ainsi, "VEHIC" et "ULE" assemblés selon A$ = "VEHIC» + "ULE" équivaut à A$ = "VEHICULE".

INSTRUCTION CURSOR

L'instruction CURSOR permet d'afficher un caractère quelconque en un point de l'écran. On accédera parfois à une fausse "haute résolution" en redéfinissant ce caractère au moyen de PATTERN.

Rappelons que CURSOR travaille avec une résolution 24 × 4Ø, l'écran étant ainsi divisé en 96Ø points.



La syntaxe d'utilisation de ces instructions est la suivante :

CURSOR (n° de colonne, n° de ligne)
où : n° de colonne est compris entre Ø et 39
: n° de ligne est compris entre Ø et 23

Cette instruction place le curseur à l'endroit spécifié sur l'écran.
En la faisant suivre d'un PRINT, on affiche un message à un endroit quelconque de l'écran.

Ainsi CURSOR 15, 1Ø : PRINT "T"
affiche la lettre "T" sur la 16e ligne, à la 11e colonne.

INSTRUCTION VPEEK

Cette puissante instruction du Basic YENO permet de "récupérer" le code ASCII d'un caractère affiché sur l'écran. Ainsi, CODE = VPEEK (40 * L + C + 15362) met dans la variable CODE la valeur ASCII du caractère de la (L + 1) ième ligne, et de la (C + 1)ième colonne (L et C définis de la même façon que pour l'instruction CURSOR).

Losanges

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : interlude.
Difficulté du jeu : aucune.

LE JEU :
Le programme va construire progressivement sur l'écran un losange noir, puis il va l'effacer progressivement, le reconstruire, et ainsi de suite.
Vous pourrez envisager de l'utiliser pour vous endormir, si les moutons ne vous suffisent plus.
Une version, très légèrement modifiée, vous est également proposée, qui vous permettra de faire apparaître des motifs aléatoires après la construction des losanges noirs.

LE PROGRAMME :
Une première boucle K (3Ø-11Ø) définit les dimensions des tracés successifs.
La deuxième boucle I (4Ø-1ØØ) trace les bandes de points, en utilisant les instructions CURSOR (lignes 6Ø à 9Ø).
Si la variable S (ligne 5Ø) est à 1, alors il y aura effacement par utilisation des instructions CURSOR avec le caractère d'Espace "" (lignes 14Ø à 17Ø).Pour obtenir un effacement aléatoire (deuxième programme), il suffira de rajouter la ligne 6Ø : 6Ø ON INT (1 + 4* RND) GOTO 14Ø, 15Ø, 16Ø, 17Ø.
On obtient alors un effacement aléatoire.

POUR JOUER :
On commence l'exécution du programme avec RUN.
Il suffit, dès lors, d'attendre et de regarder.
On pourra arrêter le déroulement à tout moment en tapant BREAK.

EXTENSIONS POSSIBLES :
On pourra immédiatement, bien sûr, songer à introduire la couleur et, éventuellement, les sons.
On peut, à partir de là, imaginer toute une série de programmes fonctionnant sur le même principe (kaléidoscope, tapisseries, etc.).

LOSANGES

```
10 CLS
20 S=0
30 FOR K=0 TO 15
40 FOR I=0 TO K
50 IF S<>0 THEN 140
60 CURSOR 16+(I-K)/2,13+I/2 :PRINT CHR
$(229)
70 CURSOR 16+(I-K)/2,13-I/2:PRINT CHR$
(229)
80 CURSOR 16+(K-I)/2,13-I/2:PRINT CHR$
(229)
90 CURSOR 16+(K-I)/2,13+I/2:PRINT CHR$
(229)
100 NEXT I
110 NEXT K
120 S=1-S
130 GOTO30
140 CURSOR 16+(I-K)/2,13+I/2:PRINT " "

150 CURSOR16+(I-K)/2,13-I/2:PRINT " "
160 CURSOR 16+(K-I)/2,13-I/2:PRINT " "

170 CURSOR 16+(K-I)/2,13+I/2:PRINT " "

180 GOTO 100
```

LOSANGES (BIS)

```
10 CLS
20 S=0
30 FOR K=0 TO 15
40 FOR I=0 TO K
50 IF S=0 THEN 70
60 ON INT(A+4*RND(1)) GOTO 140,150,160
,170
70 CURSOR 16+(I-K)/2,13+I/2:PRINT CHR$
(229)
80 CURSOR 16+(I-K)/2,13-I/2:PRINT CHR$
(229)
90 CURSOR 16+(K-I)/2,13-I/2:PRINT CHR$
(229)
100 CURSOR 16+(K-I)/2,13+I/2:PRINT CHR
$(229)
110 NEXT I
120 NEXT K
130 S=1-S
140 GOTO 30
150 CURSOR 16+(I-K)/2,13+I/2:PRINT" "
160 CURSOR 16+(I-K)/2,13-I/2:PRINT " "

170 CURSOR 16+(K-I)/2,13-I/2:PRINT " "

180 CURSOR 16+(K-I)/2,13+I/2:PRINT " "

190 GOTO110
```

Echiquier

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : graphisme.
Difficulté du jeu : variable, selon utilisation.

LE JEU :
Le programme proposé vient en appoint du programme Echecs. Il donne, en effet, la possibilité de réaliser un affichage graphique de bonne qualité pour des jeux utilisant un damier de 8 cases sur 8 cases (on pourra facilement l'étendre à 1Ø cases sur 1Ø cases). Un repérage a été prévu, horizontalement, par des lettres (de A à H), et, verticalement, par des nombres (de 1 à 8).

LE PROGRAMME :
Le code ASCII 229 définit un "pavé plein" qui sera utilisé pour dessiner les cases noires.
La boucle L (2Ø-1ØØ) parcourt les lignes de l'échiquier, la boucle C (3Ø-9Ø) les colonnes.
La ligne 4Ø affiche les lettres, la ligne 5Ø les chiffres.
La ligne 6Ø détermine (par parité) si l'on doit imprimer une case blanche ou une case noire.

POUR JOUER :
L'ordre RUN provoque l'affichage du damier sur l'écran.
Pour utiliser le programme, il conviendra de tenir compte du repérage des cases utilisables qui sont composées de 4 pavés côte à côte. Les cases sont obtenues en largeur par une tabulation de la forme 2* C et 2*L en hauteur.

EXTENSIONS POSSIBLES :
On pourra utiliser ce programme pour toutes sortes de jeux se déroulant sur un damier (échecs, dames anglaises, le loup et les agneaux, etc.).

ECHIQUIER

```
10 CLS
20 FOR L=3 TO 10
30 FOR C=5 TO 12
40 CURSOR 2*C,3:PRINT CHR$(60+C)
50 CURSOR 7,2*L:PRINT CHR$(46+L)
60 IF (C+L)/2=INT((C+L)/2) THEN 90
70 CURSOR 2*C,2*L:PRINT CHR$(229);CHR$
(229)
80 CURSOR 2*C,2*L+1:PRINT CHR$(229);CH
R$(229)
90 NEXT C
100 NEXT L
110 PRINT :PRINT
120 GOTO 120
```

Couloirs

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion.
Difficulté du jeu : difficile.

LE JEU :
Il fait appel à des notions de logique élémentaire. Le joueur peut se présenter à l'entrée, soit du couloir numéro 1, soit du couloir numéro 2. Dans ces deux entrées sont disposés des tourniquets qui basculent à votre passage et vous laissent passer, une fois vers la droite, et la fois suivante vers la gauche. De plus, un tourniquet analogue a été disposé dans le couloir central, pourrez-vous prévoir où (sortie A ou B) va s'effectuer votre prochaine sortie ?

LE PROGRAMME :
Les lignes 6Ø à 14Ø dessinent les couloirs et les trois tourniquets.
Les lignes 15Ø à 18Ø impriment les noms des entrées (1 et 2) et des sorties (A et B).
Les lignes 22Ø et 23Ø attendent votre choix.
Le décodage du chemin est fait par le YENO en utilisant trois variables logiques I, J et H, représentant les états des trois tourniquets (lignes 27Ø à 39Ø).

POUR JOUER :
Après la frappe RUN, le couloir est dessiné sur l'écran, et l'on devra taper 1 ou 2 en réponse à la question du YENO.
Dans la version retenue ici, la réponse est alors immédiatement affichée, et le chemin suivi depuis le début du jeu, ainsi que les sorties correspondantes, sont affichés sur l'écran. Ceci présente l'avantage de permettre à plusieurs joueurs d'exercer leur sagacité en essayant de prévoir, pour une entrée donnée, ce que va répondre le YENO.

EXTENSIONS POSSIBLES :
Elles seront très nombreuses ; il est très facile d'imaginer un couloir plus complexe avec d'autres entrées et d'autres sorties, et d'autres tourniquets. On peut imaginer un affichage graphique des résultats en dessinant, un bref instant, l'état des tourniquets après la réponse du joueur. On peut aussi ajouter un test des réponses (jeu à plusieurs), décompter les points de chaque joueur, etc.

COULOIRS

```
10 CLS
20 I=0
30 J=0
40 H=0
50 PRINT :PRINT ,"ENTREE SORTIE"
60 FOR K=1TO 20:CURSOR 5,2+K:PRINT CHR
$(226):NEXT K
70 FOR K=1 TO 20:CURSOR 13,2+K:PRINT C
HR$(226):NEXT K
80 FOR K=1 TO 10:CURSOR 9,2+K:PRINT CH
R$(226):NEXT K
90 FOR K= 1TO 4:CURSOR 9,18+K:PRINT CH
R$(226):NEXT K
100 FOR K= 1TO 7:CURSOR 7,10+ K:PRINT
CHR$(226):NEXT K
110 FOR K= 1TO 7:CURSOR 11,10+ K:PRINT
 CHR$(226):NEXT K
120 CURSOR 9,14:PRINT "T"
130 CURSOR 7,20:PRINT "T"
140 CURSOR 11,20:PRINT "T"
150 CURSOR 11,3:PRINT "B"
160 CURSOR 7,3:PRINT "A"
170 CURSOR 7,23:PRINT "1"
180 CURSOR 11,23:PRINT "2"
190 L=6
200 CURSOR 19,L:PRINT ">"
210 BEEP
220 K$=INKEY$
230 IF K$<>"1" AND K$<>"2" THEN 220
240 CURSOR 19,L:PRINT " "
250 CURSOR 20,L:PRINT K$
260 IF K$="1" THEN 310
270 J=1-J
280 IF J<>0 THEN 370
290 O$="B"
300 GOTO 340
310 I=1-I
320 IF I<>0 THEN 370
330 O$= "A"
340 CURSOR 28,L:PRINT O$
350 L=L+1
360 GOTO 200
370 H=1-H
380 IF H<>0 THEN 290
390 GOTO 340
```

Jour de la semaine

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : utilitaire.
Difficulté du jeu : facile.

LE JEU :
Le programme vous permet de retrouver, sans aucun calcul, le jour de la semaine correspondant à une date donnée.
La date sera entrée sous forme de trois nombres, par exemple 11, 3, 1956 pour le 11 mars 1956, et le résultat sera affiché en clair (sous forme un peu abrégée).

LE PROGRAMME :
Les lignes 1Ø à 4Ø contrôlent les entrées. La ligne 2Ø s'occupe de la fin de partie. La ligne 8Ø permet une correction pour les dates antérieures à 1582 (mise en service du calendrier actuel).
Les lignes 9Ø à 1ØØ permettront l'affichage en clair du mois dans la réponse (sur 4 lettres).
La formule utilisée apparaît aux lignes 13Ø et 14Ø.
Les lignes 15Ø et 16Ø permettront l'affichage en clair du jour dans la réponse. La ligne 17Ø imprime le résultat.
Le sous-programme 19Ø-21Ø intervient pour tenir compte des mois de janvier et février.

POUR JOUER :
Lancer le programme par RUN, puis entrer le jour suivi de CR, le mois (entre 1 et 12) suivi de CR, et l'année (sous la forme 1845, par exemple) suivie de CR.
Le résultat est très rapidement affiché, et le programme revient sur la question JOUR.
Pour continuer, il suffit de retaper un jour, un mois, une année.
Pour terminer, il faut répondre Ø à la question JOUR.

EXTENSIONS POSSIBLES :
On pourra entrer les mois en clair et faire afficher le nom du jour en entier.
On peut envisager de faire afficher les dates et fêtes particulières (Noël, Pâques, les Rameaux, Pentecôte...).
On peut écrire un programme qui afficherait les jours fériés pour une année donnée, etc.

JOUR DE LA SEMAINE

```
10 INPUT "JOUR       :";J
20 IF J=0 THEN 220
30 INPUT "MOIS(1-12):";M
40 INPUT "ANNEE      :";A
50 B=A
60 S=INT(A/100)-INT(A/400)
70 IF A>1582 THEN 90
80 S=2
90 M$="JANVFEVRMARSAVRIMAI JUINJUILAOU
SEPTOCTONOVEDECE"
100 M$=MID$(M$,4*M-3,4)
110 IF M>=3 THEN 130
120 GOSUB 190
130 S=J-1+INT(13*(M+1)/5)+INT(5*A/4)-S

140 S=S-7*INT(S/7)
150 J$="DIMANCHELUNDI   MARDI   MERCRE
DIJEUDI   VENDREDISAMEDI  "
160 J$=MID$(J$,8*S+1,8)
170 PRINT :PRINT "LE ";J;M$;B;" ETAIT
UN ";J$:PRINT
180 GOTO 10
190 M=M+12
200 A=A-1
210 RETURN
220 END
```

Le pendu

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de lettres (version deux joueurs).
Difficulté du jeu : moyenne.

LE JEU :
Il serait sans doute assez difficile de retrouver le nom de l'inventeur de ce jeu très à la mode, mais il est certain qu'il existe depuis des temps immémoriaux, quelquefois pratiqué sous des formes assez différentes (par exemple, reconstituer un petit cochonnet en lançant des dés).
L'apparition des jeux sur micro-ordinateur et la magie des écrans de télévision lui ont fait connaître un renouveau certain : dans la version retenue, l'un des joueurs doit entrer un mot de longueur quelconque. L'ordinateur affiche alors des tirets à l'emplacement des lettres de ce mot et demande au deuxième joueur de proposer une lettre. Chaque réponse correcte fait apparaître la lettre à son emplacement exact dans le mot, et chaque réponse fausse fait progresser le dessin de la potence.
Pour gagner, il suffit de réussir à trouver le mot inconnu avant d'être pendu.

LE PROGRAMME :
Le mot à deviner est introduit en début de programme (ligne 3Ø) ; une variable B$, en ligne 2Ø contient tous les éléments du dessin.
La boucle I (5Ø-7Ø) fabrique un mot P$ qui contient des tirets à l'emplacement de chacune des lettres du mot initial.
La boucle I (11Ø-13Ø) s'occupe du dessin (plus ou moins complet, selon la valeur de P).
Les lignes 16Ø et 19Ø affichent le résultat et arrêtent le jeu.
La lettre essayée est entrée en ligne 21Ø et testée dans une boucle I (23Ø-26Ø) ; le sous-programme 3ØØ-32Ø remplace chaque tiret par la lettre correspondante.

POUR JOUER :
Après le lancement du programme par RUN, l'un des joueurs doit entrer son mot et taper CR. Des tirets sont affichés à l'emplacement de chacune des lettres du mot ; et la question LETTRE apparaît sur l'écran ; l'autre joueur répond alors en proposant une lettre suivie de CR.
Si la lettre est bonne, elle viendra remplacer un ou plusieurs tirets ; sinon, la potence va commencer à se dessiner.
Le mot exact sera affiché dans le cas où l'on aurait perdu.

EXTENSIONS POSSIBLES :
Il sera intéressant de constituer une liste des mots dans laquelle YENO pourra piocher au hasard ; on pourra aussi améliorer le dessin et modifier le nombre d'essais conduisant à la pendaison, en fonction de la difficulté du mot.

LE PENDU

```
10 CLS
20 B$="        I II I888HHHHHHHHH I A ¦
 ¦ ---"
30 INPUT "VOTRE MOT :";R$
40 P$=""
50 FOR I=1 TO LEN(R$)
60 P$=P$+"="
70 NEXT I
80 P=1
90 CLS
100 PRINT "LE MOT: ";P$
110 FOR I=0 TO P
120 PRINT TAB(20);MID$(B$,1+3*I,3)
130 NEXT I
140 PRINT :PRINT :PRINT :PRINT
150 IF P$<>R$ THEN 180
160 PRINT ,"BRAVO"
170 GOTO 330
180 IF P<>13 THEN 210
190 PRINT "LE MOT ETAIT ";R$
200 GOTO 330
210 INPUT "VOTRE LETTRE ? ";S$
220 T=0
230 FOR I=1 TO LEN(R$)
240 IF MID$(R$,I,1)<>S$ THEN 260
250 GOSUB 300
260 NEXT I
270 IF T<>0 THEN 90
280 P=P+1
290 GOTO 90
300 T=1
310 P$=LEFT$(P$,I-1)+S$+RIGHT$(P$,LEN(
P$)-I)
320 RETURN
330 END
```

Bridge

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion.
Difficulté du jeu : variable, selon le problème.

LE JEU :
Vous ne trouverez pas, dans ce programme, un partenaire pour vos bridges mais, plus exactement, une simulation pour l'affichage d'une donne ; cet affichage est réalisé jeu par jeu. On peut, à partir de là, se poser toutes sortes de problèmes : annonces, jeu de la carte, etc.

LE PROGRAMME :
Les lignes 3Ø et 4Ø définissent un jeu de 52 cartes.
La variable B$, en ligne 6Ø, permettra de donner en abrégé le nom des couleurs et aussi d'améliorer la désignation de certaines cartes (1Ø, valet, dame, roi et as).
La boucle J (9Ø-22Ø) réalise le tirage de quatre jeux de 13 cartes.
La boucle I (8Ø-23Ø) réalise tirage et impression. La carte est choisie parmi les 52 disponibles, en ligne 1ØØ.
La ligne 11Ø calcule la couleur de la carte tirée.
La ligne 12Ø vérifie que cette carte n'a pas déjà été tirée.
Les lignes 15Ø et 16Ø s'occupent de l'impression complémentaire dans le cas où l'on tire un 1Ø, un valet, une dame, un roi ou un as.
L'impression est réalisée par les lignes 14Ø et 17Ø.
La ligne 18Ø permet d'éliminer les cartes déjà tirées.

POUR JOUER :
On lance le programme par RUN.
Quatre séries de 13 cartes sont affichées verticalement.

EXTENSIONS POSSIBLES :
Sur un matériel disposant, comme c'est le cas du YENO, d'un affichage performant, on peut imaginer de représenter les 13 cartes sous une forme plus conventionnelle.
On peut aussi ajouter la possibilité de faire des annonces (et de les comparer à celles du YENO, par exemple), ou introduire des possibilités de jeu de la carte, y compris pour le YENO.

BRIDGE

```
10 CLS
20 H=0
30 R$="234567891VDRA"
40 R$=R$+R$+R$+R$
50 U=7
60 B$="TRCACOPI0AAOS"
70 V=LEN(B$)
80 FOR I=1 TO V
90 FOR J=1 TO 4
100 X=INT(LEN(R$)*RND(1)+1)
110 S=2*INT((X-1)/V)+1
120 IF MID$(R$,X,1)="0" THEN 100
130 T=X-V*INT((X-1)/V)
140 PRINT MID$(R$,X,1);
150 IF T<9 THEN PRINT " ";:GOTO 170
160 PRINT MID$(B$,S,2);" ";
170 PRINT " ";MID$(B$,S,2);" ";
180 R$=LEFT$(R$,X-1)+"0"+RIGHT$(R$,LEN
(R$)-X)
190 H=H+1
200 IF H=5 THEN H=0:PRINT :PRINT
210 PRINT TAB(7*H);
220 NEXT J
230 NEXT I
```

Horloge 1

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : dessin.
Difficulté du jeu : aucune.

LE JEU :
Transformer votre écran de télévision en horloge numérique à affichage grand format, grâce à ce programme qui vous permettra d'animer une vitrine, de faire une horloge fantaisie, et qui pourra se prêter à bien d'autres usages.

LE PROGRAMME :
La ligne 3Ø permet d'entrer l'heure sous forme de 3 entiers de 2 chiffres : 1Ø:24:Ø2 signifie 1Ø heures 24 minutes et 2 secondes.
Les variables B$ et C$ (4Ø-5Ø) codent le dessin des chiffres. La ligne 6Ø règle l'horloge interne du YENO. La ligne 8Ø effectue le décodage de chaque chiffre.
La boucle I (9Ø-17Ø) assure l'impression de chaque chiffre, au fur et à mesure qu'il change (test 1ØØ). La boucle J (12Ø-16Ø) assure le décodage de B$ (13Ø) et affiche le dessin du chiffre (14Ø-15Ø).

POUR JOUER :
Dès le lancement par RUN, le programme se place en attente de l'heure ; on tape au prochain top sonore et on attend le top pour appuyer sur CR. L'heure s'affiche alors assez rapidement et l'horloge est en route. Pour l'arrêter, il faudra appuyer sur la touche BREAK.
L'heure est entrée sous forme de 3 entiers ayant impérativement 2 chiffres, pour 5 heures 6 minutes 20 secondes, il faut taper Ø5:Ø6:2Ø puis CR.

EXTENSIONS POSSIBLES :
Il sera intéressant de modifier le programme pour en faire un réveil, avec alarme ; on peut aussi améliorer le graphisme, introduire la couleur.

HORLOGE 1

```
10 CLS
20 FOR I=1 TO 4:H(I)=10:NEXT I
30 INPUT "HEURE (format: HH:MM:SS)  :"
;A$
40 B$="144413333131211313144133121311
2141133331414114131"
50 C$=CHR$(229)+CHR$(229)+CHR$(224)+"
 "+CHR$(227)+CHR$(224)+CHR$(227)
60 TIME$=A$
70 CLS
80 A$=LEFT$(TIME$,2)+MID$(TIME$,4,2)
90 FOR I=1 TO 4
100 IF H(I)=VAL(MID$(A$,I,1)) THEN 170
RUN
110 H(I)=VAL(MID$(A$,I,1))
120 FOR J=1 TO 5
130 S=2*VAL(MID$(B$,J+5*H(I),1))-1
140 CURSOR 7+3*I+3,10+J:PRINT MID$(C$,
S,1)
150 CURSOR 7+3*I+4,10+J:PRINT MID$(C$,
S+1,1)
160 NEXT J
170 NEXT I
180 GOTO 80
```

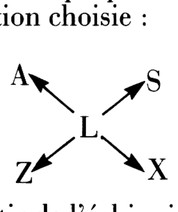
Le loup et les agneaux

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : moyenne.

LE JEU :
C'est un jeu assez connu qui peut être pratiqué indifféremment sur un damier 8 sur 8, ou 1Ø sur 1Ø. Il est présenté, ici, dans une version pour un damier de 8 cases ; l'un des joueurs choisit le loup qui se déplace suivant les diagonales, en restant sur les cases noires, d'une case à chaque coup, et qui peut avancer ou reculer. Les agneaux se déplacent également sur les cases noires, en diagonale, mais n'ont pas le droit de reculer. Les cases déjà occupées sont interdites (pas de prises), et il n'y a pas de saut.
Le loup gagne s'il réussit à arriver sur la ligne du bas avant que tous les agneaux n'occupent celle du haut ; les agneaux gagnent dans ce dernier cas, ou s'ils arrivent à bloquer le loup.

LE PROGRAMME :
Une variable A$, en ligne 6Ø contient les positions de départ.
Les boucles L (8Ø-13Ø) et C (9Ø-12Ø) dessinent le damier, et les lignes 14Ø à 18Ø placent le loup et les agneaux.
La ligne 19Ø gère le changement de joueur.
La ligne 2ØØ affiche L si c'est le joueur qui manipule le loup qui a le trait, et A si c'est son adversaire ; la ligne 21Ø attend le coup du joueur.
La ligne 22Ø choisira le programme analysant la réponse, en fonction du joueur concerné.

POUR JOUER :
RUN déclenche l'arrêt sur l'attente d'une entrée : Ø et CR si l'on veut que les agneaux commencent, et 1 pour le loup. Après quoi, le damier s'affiche et la phrase COUP A pour les agneaux, et COUP L pour le jeu du loup.
Les agneaux répondent par un numéro, suivi de D ou G (pour droite ou gauche).
Ainsi 2D provoque le déplacement de l'agneau numéro 2 qui monte en diagonale d'une case sur la droite. Le déplacement effectué est affiché, puis on obtient une question identique pour le loup qui répondra en tapant A, S, X ou Z, suivant la direction choisie :



On devra veiller à ne pas sortir de l'échiquier et à ne pas venir sur l'adversaire, aucun contrôle n'ayant été prévu.

EXTENSIONS POSSIBLES :
On pourra prévoir un repérage des cases, un graphisme plus soigné, des contrôles de non dépassement des limites du terrain, de non venue sur une case déjà occupée...
On pourra aussi envisager de jouer sur un damier de 10 cases sur 10 cases, ce qui augmente légèrement la difficulté du jeu pour les agneaux. Mais l'extension la plus intéressante à réaliser sera certainement la réalisation d'un programme permettant au YENO de se comporter en adversaire honorable, soit avec le loup, soit avec les agneaux (et, dans ce cas, il sera possible de définir une stratégie gagnante pour le YENO).

```
        LE LOUP ET LES AGNEAUX


10 CLS
20 DIM B$(1)
30 B$(1)="A"
40 B$(0)="L"
50 INPUT "AGNEAUX (0) OU LOUP (1) ? ";
J
60 A$="61636567714"
70 CLS
80 FOR L=8 TO 15
90 FOR C=10 TO 17
100 IF (L+C)/2=INT((L+C)/2) THEN 120
110 CURSOR C,L:PRINT CHR$(144)
120 NEXT C
130 NEXT L
140 CURSOR VAL(MID$(A$,10,1))+9,VAL(MI
D$(A$,9,1))+7:PRINT "L"
150 CURSOR VAL(MID$(A$,2,1))+9,VAL(MID
$(A$,1,1))+7:PRINT "1"
160 CURSOR VAL(MID$(A$,4,1))+9,VAL(MID
$(A$,3,1))+7:PRINT "2"
170 CURSOR VAL(MID$(A$,6,1))+9,VAL(MID
$(A$,5,1))+7:PRINT "3"
180 CURSOR VAL(MID$(A$,8,1))+9,VAL(MID
$(A$,7,1))+7:PRINT "4"
190 J=ABS(J-1)
200 CURSOR 0,20:PRINT "COUP ";B$(J);"
";
210 INPUT R$
220 IF J<>0 THEN 260
230 C=10
240 A=VAL(MID$(A$,C-1,10))-(R$="X")*(C
+1)-(R$="Z")*(C-1)+(R$="A")*(C-1)+(R$=
"S")*(C-1)
250 GOTO 280
260 C=2*VAL(MID$(R$,1,1))
270 A=VAL(MID$(A$,C-1,2))+9*(MID$(R$,2
,1)="D")+11*(MID$(R$,2,1)="G")
280 A$=MID$(A$,1,C-2)+RIGHT$(STR$(A),2
)+MID$(A$,C+1,10)
290 GOTO 70
```

Belote

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de cartes.
Difficulté du jeu : facile.

LE JEU :
Le jeu proposé est plutôt une simulation, réalisée sur 32 cartes, de la distribution de 4 paquets de 5 cartes. Il pourra être utilisé pour l'étude de problèmes faisant intervenir des donnes de 2Ø cartes sur 32.

LE PROGRAMME :
En ligne 2Ø, on crée un tableau pouvant contenir les 2Ø cartes.
La boucle I (4Ø-1ØØ) assure le tirage aléatoire de 2Ø nombres, pris parmi 32 ; la boucle J (7Ø-9Ø) vérifie que l'on ne réalise pas deux fois le même tirage.
La deuxième boucle I (11Ø-35Ø) assure ensuite l'affichage, les noms des cartes étant affichés en clair, et la couleur étant décodée et affichée en clair, grâce aux lignes 17Ø, 19Ø, 21Ø et 23Ø.

POUR JOUER :
Il suffit de taper RUN pour obtenir immédiatement l'affichage de 4 paquets de 5 cartes chacun (une donne).
Un nouveau RUN permet d'obtenir une nouvelle distribution.

EXTENSIONS POSSIBLES :
On pourra facilement mettre au point un affichage plus graphique ; on pourra également se servir de ce programme comme d'un noyau pour la mise au point de programmes de jeu de cartes sur l'ordinateur (qui pourra, si nécessaire, servir de partenaire : voir l'exemple du Poker, dans le même ouvrage).

BELOTE

```
10 CLS
20 DIM A(20)
30 K=1
40 FOR I=1 TO 20
50 A(I)=1+INT(32*RND(1))
60 IF I=1  THEN 100
70 FOR J=1 TO I-1
80 IF A(I)=A(J) THEN 50
90 NEXT J
100 NEXT I
110 FOR I=1 TO 20
120 S$="CA"
130 S=1+A(I)-8*INT(A(I)/8)
140 R$=STR$(S+6)
150 R$=MID$(R$,2,2)
160 IF S<>5 THEN 180
170 R$="VA"
180 IF S<>6 THEN 200
190 R$="DA"
200 IF S<>7 THEN 220
210 R$="RO"
220 IF S<>8 THEN 240
230 R$="AS"
240 IF A(I)>24 THEN 260
250 S$="TR"
260 IF A(I)>16 THEN 280
270 S$="PI"
280 IF A(I)>8 THEN 300
290 S$="CO"
300 PRINT TAB(7*K-6);R$;TAB(7*K-3);S$;

310 K=K+1
320 IF K<>6 THEN 350
330 K=1
340 PRINT :PRINT
350 NEXT I
```

Un taquin de calcul

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : difficile.

LE JEU :
Il se pratique à deux joueurs, l'un tenant le rôle du codeur, l'autre celui du décodeur.
Le codeur part d'un nombre (proposé par le YENO) et doit le transformer à l'aide de 5 opérations qu'il choisit lui-même, ainsi d'ailleurs que les opérateurs.
Le résultat qu'il a obtenu et les opérations qu'il a utilisées seront affichés sur l'écran : le décodeur devra alors essayer de retrouver le nombre initial en effectuant les opérations en sens inverse.

LE PROGRAMME :
La ligne 3Ø permet d'obtenir l'entier de départ compris entre 5 et 2Ø.
Une boucle I (9Ø-17Ø) permet d'entrer successivement les 5 opérations.
4 sous-programmes sont utilisés pour les calculs (aux lignes 23Ø à 3ØØ).
Après chaque calcul, le résultat est affiché (seul, le codeur devra regarder l'écran pendant cette opération) et le résultat définitif restera affiché pour le décodeur à la fin des calculs (ligne 16Ø).
La ligne 19Ø affiche les 5 opérations utilisées, ainsi que les opérateurs correspondants.

POUR JOUER :
On frappe RUN pour lancer le programme.
Le nombre de début est affiché, puis la question ENTREZ 5 OPERATIONS ; on devra les entrer l'une après l'autre, sous la forme, par exemple : + 4 suivi de CR.
Le résultat sera affiché après chaque opération entrée. A la fin, les opérations entrées sont toutes affichées sur une même ligne, et le message BON COURAGE est envoyé à l'adresse du décodeur (il n'y a pas de contrôle au décodage).

EXTENSIONS POSSIBLES :
On pourra, bien sûr, songer à améliorer la partie décodage du programme en introduisant des contrôles sur le résultat ; on peut aussi envisager de laisser le YENO participer davantage en lui confiant le rôle du codage, etc.

UN TAQUN DE CALCUL

```
10 DIM R$(5)
20 CLS
30 X=INT(16*RND(1)+5)
40 PRINT "NOMBRE INITIAL :";X
50 L=1
60 PRINT "ENTREZ 5 OPERATIONS "
70 F=X
80 PRINT "FORME +6,/5......"
90 FOR I=L TO 5
100 INPUT R$(I)
110 A=ASC(MID$(R$(I),L,1))-41
120 IF (A<=0) OR (A=3) OR (A=5) OR (A>
6)  THEN 200
130 IF A<=3 THEN 150
140 A=A/2+1
150 ON A GOSUB 230,250,270,290
160 PRINT "NOMBRE OBTENU :";F
170 NEXT I
180 CLS
190 PRINT :PRINT "NOMBRE OBTENU :";F:P
RINT :PRINT "OPERATIONS ":PRINT R$(4);
TAB(5);R$(1);TAB(10);R$(5);TAB(15);R$(
3);TAB(20);R$(2):PRINT "BON COURAGE"
200 K$=INKEY$
210 IF K$="" THEN 200
220 GOTO 20
230 F=F*VAL(MID$(R$(I),2,3))
240 RETURN
250 F=F+VAL(MID$(R$(I),2,3))
260 RETURN
270 F=F-VAL(MID$(R$(I),2,3))
280 RETURN
290 F=F/VAL(MID$(R$(I),2,3))
300 RETURN
```

Poker

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu contre le YENO.
Difficulté du jeu : facile.

LE JEU :
C'est une simulation d'une partie de poker traditionnelle, effectuée avec un jeu de 32 cartes, dans laquelle le joueur aura, pour partenaire, le YENO.
Le joueur recevra 5 cartes, repérées par des numéros de 1 à 5, et pourra décider de changer certaines cartes en annonçant celles qu'il écarte. Sa nouvelle donne sera alors affichée, et c'est ensuite au tour du YENO de jouer.

LE PROGRAMME :
Les lignes 4Ø et 5Ø définissent les cartes. Une variable N est dimensionnée à 15, en ligne 1Ø, pour recevoir 1Ø cartes joueur et 5 cartes YENO.
Le tirage sera effectué par une boucle I (7Ø-12Ø) et le contrôle par une boucle J (9Ø-11Ø).
L'affichage des cartes est réalisé par le sous-programme 23Ø-32Ø.
Une autre boucle I (18Ø-2ØØ) réalise le deuxième tirage du joueur, la ligne 21Ø permettant le deuxième affichage, et la boucle I (23Ø-28Ø) où l'on entre directement après, réalisera l'affichage du jeu machine.

POUR JOUER :
L'ordre RUN lance l'exécution du programme, et les 5 premières cartes du joueur sont alors affichées.
A la question CARTES REFUSEES, le joueur peut répondre Ø suivi de CR s'il veut arrêter la partie. Pour jouer, il répondra simplement CR s'il s'estime servi, ou il entrera les numéros des cartes qu'il refuse suivis de CR.
Par exemple, s'il tape 235 et CR, son nouveau jeu conservera les cartes 1 et 4. Son jeu sera alors affiché ainsi que celui du YENO, ce dernier n'ayant pas, dans ce programme, la possibilité de reprendre des cartes.
Il suffira ensuite de faire CR pour démarrer une autre partie.

EXTENSIONS POSSIBLES :
Elles seront, bien sûr, très nombreuses, depuis un affichage graphique en couleur des cartes jusqu'à l'amélioration du jeu du YENO, l'introduction des règles sur les paris et même, pourquoi pas, des possibilités de bluff.

POKER

```
10 DIM N(15)
20 CLS
30 S=6
40 V$="7 8 9 10VADAROAS"
50 C$="TRCACOPI"
60 N(1)=INT(32*RND(1))
70 FOR I=2 TO 15
80 N(I)=INT(32*RND(1))
90 FOR J=1 TO I-1
100 IF N(I)=N(J) THEN 80
110 NEXT J
120 NEXT I
130 K=0
140 PRINT "CARTES REFUSEES"
150 GOSUB 240
160 IFR$<>"0" THEN 180
170 END
180 FOR I=1 TO LEN(R$)
190 N(VAL(MID$(R$,I,1)))=N(I+S-1)
200 NEXT I
210 CLS
220 GOSUB 240
230 K=10
240 FOR I=1 TO 5
250 J=2*(N(I+K)-4*INT(N(I+K)/4)+1)-1
260 F=2*(N(I+K)-7*INT(N(I+K)/7)+1)-1
270 R$=MID$(V$,F,2)
280 PRINT TAB(S*I-S);I;":";TAB(S*I-S);
R$;"/";MID$(C$,J,2)
290 NEXT I
300 INPUT R$
310 IF K<>0 THEN 20
320 RETURN
```

Hanoï

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion.
Difficulté du jeu : difficile.

LE JEU :
Il est présenté, ici, dans une version pour un joueur. On dispose d'un certain nombre de disques (8 au plus) et l'on doit les transférer de la tour numéro 1 à la tour numéro 3, en respectant une seule règle : à aucun moment, un disque ne peut être recouvert par un disque d'un diamètre supérieur.
Le nombre de coups n'est pas comptabilisé par ce programme mais, par contre, les coups illicites seront refusés.

LE PROGRAMME :
Le caractère 229, en ligne 23Ø, servira à dessiner les disques.
Le nombre de disques est entré en ligne 7Ø et refusé s'il y a plus de 8 disques.
La boucle I (9Ø-11Ø) permet de fabriquer une variable A (I) qui contient l'état de la pile sous forme numérique (en ligne 1Ø, trois variables numériques sont prévues pour conserver l'état des trois piles).
Les socles sont dessinés par les lignes 13Ø à 15Ø et les trois piles par les boucles emboîtées I (19Ø-25Ø) et J (2ØØ-24Ø).
Le contrôle du coup est effectué par la ligne 31Ø et l'état des piles est modifié par les lignes 32Ø et 33Ø.

POUR JOUER :
Après RUN qui lance le programme, on entre le nombre de disques demandés, suivi de CR. On obtient le dessin de l'état initial et, à la question TOUR DE DEPART, on répond en donnant le numéro de la tour de départ. Ce numéro (1, 2 ou 3) sera refusé si la tour est vide. Puis on tape CR et, à la question TOUR D'ARRIVEE, on répond par le numéro de la tour d'arrivée.
Si le coup n'est pas possible, on sera obligé de recommencer à TOUR DE DEPART. Si le coup est possible, le déplacement est réalisé et l'on recommence.

EXTENSIONS POSSIBLES :
On pourra dessiner des disques de différentes couleurs, introduire un compteur de coups et, surtout, définir un jeu pour l'ordinateur.
On peut aussi essayer de compliquer le jeu avec deux tiges de plus et deux joueurs, chacun ayant des disques d'une couleur différente, par exemple.

HANOI

```
10 DIM A(3)
20 CLS
30 U=9
40 A(1)=U
50 A(2)=U
60 A(3)=U
70 INPUT "NOMBRES DE DISQUES :";N
80 IF N>7 THEN 20
90 FOR I=0 TO N-1
100 A(1)=VAL(STR$(A(1))+STR$(N-I))
110 NEXT I
120 CLS
130 CURSOR 3,15:PRINT "^^^^^^^^^"
140 CURSOR 13,15:PRINT "^^^^^^^^^"
150 CURSOR 23,15:PRINT "^^^^^^^^^"
160 CURSOR 7,17:PRINT "1"
170 CURSOR 17,17:PRINT "2"
180 CURSOR 27,17:PRINT "3"
190 FOR I=0 TO N
200 FOR J=1 TO 3
210 B$=STR$(A(J))
220 IF LEN(B$)<=I+1 THEN 240
230 CURSOR (U+1)*J-U+2,U-I+5:FOR K=1 T
O VAL(MID$(B$,I+2,1)):PRINT CHR$(229);
:NEXT K
240 NEXT J
250 NEXT I
260 CURSOR 0,20:INPUT "TOUR DE DEPART
:";D
270 IF A(D)=U THEN 260
280 INPUT "TOUR D'ARRIVEE :";W
290 B$=STR$(A(D))
300 C$=STR$(A(W))
310 IF RIGHT$(B$,1)>RIGHT$(C$,1) THEN
260
320 A(W)=VAL(C$+RIGHT$(B$,1))
330 A(D)=VAL(LEFT$(B$,LEN(B$)-1))
340 GOTO 120
```

Musique

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu éducatif.
Difficulté du jeu : moyenne.

LE JEU :
Il se présente comme un exercice d'apprentissage du solfège : l'appareil annonce une note (en clé de sol), suivie de la mention GRAVE ou AIGUE, selon sa place sur la portée. Il faudra répondre en indiquant sur quelle ligne la note sera affichée.
La note est alors dessinée sur la portée, suivie d'un message BRAVO ou PERDU, selon que l'on avait trouvé juste ou non.

LE PROGRAMME :
La boucle I (7Ø-9Ø) dessine la portée. Les deux boucles I (1ØØ-12Ø) et (13Ø-15Ø) affichent les numéros des lignes.
La ligne 16Ø définit les noms des notes, et la ligne 17Ø réalise le tirage aléatoire d'une note.
Le nom de la note est calculé à la ligne 18Ø et il est affiché par la ligne 19Ø, suivi de la mention GRAVE ou AIGUE, selon le cas.
La réponse est entrée en ligne 2ØØ.
La ligne 21Ø s'occupe du décodage de la réponse.
Les lignes 22Ø à 24Ø affichent la note.
La ligne 25Ø regarde si l'on a gagné ou perdu et affiche le message correspondant.
Les lignes 26Ø à 28Ø permettent de faire un nouvel essai.

POUR JOUER :
On lance le programme par RUN.
On obtient alors l'affichage de la portée, ainsi que le nom de la note à placer et la question LIGNE.
On répond en donnant un numéro de ligne (de 1 à E) suivi de CR.
La note est alors dessinée et le message BRAVO ou PERDU apparaît, dans ce cas, à l'écran.
On peut taper une touche quelconque pour continuer, et s'arrêter en tapant BREAK.

EXTENSIONS POSSIBLES :
On peut étendre à une autre clé que la clé de sol, on peut introduire de la couleur, des sons, etc.

```
10 DIM T$(1),RR$(1)
20 RR$(1)="BRAVO"
30 RR$(0)="PERDU"
40 T$(1)="GRAVE"
50 T$(0)="AIGUE"
60 CLS
70 FOR I=1 TO 5
80 CURSOR 8,3+2*I:PRINT CHR$(233);CHR$
(233);CHR$(233);CHR$(233);CHR$(233)
90 NEXT I
100 FOR I=1 TO 9
110 CURSOR 3,I+1:PRINT CHR$(48+I)
120 NEXT I
130 FOR I=10 TO 14
140 CURSOR 3,I+1:PRINT CHR$(55+I)
150 NEXT I
160 N$="SI LA SOLFA MI RE DO"
170 X=INT(14*RND(1)+1)
180 T=X-1-7*INT((X-1)/7)
190 CURSOR 0,20:PRINT "NOTE ";MID$(N$,
3*T+1,3);"  ";T$(-(X>7))
200 INPUT "LIGNE :";L$
210 L=(L$>"@")*(55-ASC(L$))-(ASC(L$)-4
8)*(L$<":")
220 CURSOR 10,X-1:PRINT CHR$(227)
230 CURSOR 10,X:PRINT CHR$(227)
240 CURSOR 10,X+1:PRINT CHR$(236)
250 CURSOR 15,21:PRINT RR$(-(L=X))
260 K$=INKEY$
270 IF K$="" THEN 260
280 GOTO 60
```

Pendu inversé

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu sur les mots.
Difficulté du jeu : facile.

LE JEU :
Le Pendu est un jeu de lettres très connu, présenté par ailleurs dans ce recueil, dans lequel il s'agit de deviner un mot caché par l'ordinateur et où chaque lettre non découverte vous rapproche un peu plus de la potence. Ici, c'est le YENO qui va chercher à deviner votre mot, mais il joue quelque peu au hasard et ne possède pas votre intuition ; il faudra donc, en règle générale, beaucoup d'essais pour découvrir la solution. Pour augmenter ses chances, on a fait appel à une formule qui lui permet de tirer davantage dans les voyelles puis, dans les consonnes courantes, et, enfin, dans les consonnes plus rares.

LE PROGRAMME :
La ligne 2Ø permet d'enregistrer, dans une variable alphanumérique D$, les lettres de l'alphabet, dans l'ordre : voyelles, consonnes usuelles, autres consonnes. Le mot à deviner est introduit en ligne 5Ø (en fait, il suffirait d'introduire sa longueur).
La boucle I (6Ø-8Ø) imprime des tirets aux emplacements des lettres à trouver.
La formule, placée en ligne 1ØØ, permet de modifier légèrement la probabilité de tirage, en faveur des voyelles et des consonnes usuelles.
La ligne 12Ø provoque un nouveau tirage, si la lettre avait déjà été essayée.La ligne 15Ø affiche la lettre choisie, et la boucle I (16Ø-19Ø) réimprime les tirets et les lettres trouvées.
La ligne 2ØØ incrémente le compteur d'essais.

POUR JOUER :
Lancer le programme par RUN.
Entrer le mot à découvrir, suivi de CR.
Des tirets sont alors affichés (un par lettre) et le YENO propose une première lettre, regarde si elle convient et continue ceci automatiquement, jusqu'à ce qu'il ait trouvé.
On pourra sortir du programme à tout moment en tapant BREAK.

EXTENSIONS POSSIBLES :
Il sera sans doute intéressant, pour de jeunes enfants, de prévoir une version non automatique où le joueur répondrait OUI ou NON pour chaque lettre proposée par le YENO ; on pourra aussi imaginer d'ajouter un affichage graphique, par exemple une potence ou une voiture qui avance vers un précipice, etc.
Avec une minidisquette, on pourra faire retenir, au YENO, les mots les plus couramment utilisés par les joueurs, etc.

PENDU INVERSE

```
10 CLS
20 D$="YAEIOUBCDFGHJKLMNPQRSTVWXZ"
30 C=1
40 T$=""
50 INPUT "VOTRE MOT :";R$
60 FOR I=1 TO LEN(R$)
70 T$="="+T$
80 NEXT I
90 PRINT :PRINT T$:PRINT :PRINT :PRINT
 "JE CHERCHE.....ESSAI ";C
100 X=-6*(RND(1)<.7)-13*(RND(1)<.5)-7*
(RND(1)<.4)
110 X=INT(X*RND(1)+1)
120 IF MID$(D$,X,1)="=" THEN 100
130 E$=MID$(D$,X,1)
140 D$=LEFT$(D$,X-1)+"="+RIGHT$(D$,26-
X)
150 PRINT :PRINT "JE PRENDS ";E$
160 FOR I=1 TO LEN(R$)
170 IF E$<>MID$(R$,I,1) THEN 190
180 T$=LEFT$(T$,I-1)+LEFT$(E$,1)+RIGHT
$(T$,LEN(T$)-I)
190 NEXT I
200 C=C+1
210 IF T$<>R$ THEN 90
```

Mathieu

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : dessin aléatoire.
Difficulté du jeu : aucune.

LE JEU :
On utilise la définition d'une courbe en coordonnées polaires et la fonction HCHAR pour obtenir sur l'écran des dessins aléatoires (de type fleur), plus ou moins élégants.
Les deux paramètres principaux, A et B, pourront être, si nécessaire, entrés à la main (instruction INPUT) pour faire une étude plus systématique des différentes courbes possibles.

LE PROGRAMME :
La ligne 2Ø permet de définir le rayon maximum et la ligne 3Ø corrige ce paramètre pour utiliser, au mieux, la surface de l'écran.
On pourra, évidemment, modifier ces deux paramètres en veillant à rester dans les limites de l'écran.
Les lignes 4Ø et 5Ø définissent deux variables aléatoires A et B, choisies ici entre Ø et 3 ; l'allure des courbes dépendant de ces variables, on pourra modifier leurs intervalles de valeurs, ou même les entrer à la main, en écrivant 5Ø INPUT A, B.
La boucle I (6Ø-12Ø) réalise le tracé, point par point. Le rayon R est calculé à la ligne 7Ø, et le passage en coordonnées cartésiennes (indispensable pour CURSOR) est effectué aux lignes 8Ø et 9Ø.
La ligne 1ØØ réalise l'impression des points.
La ligne 13Ø affiche le message FINI.
Les lignes 14Ø et 15Ø permettent l'exécution d'un autre dessin, dès que l'on effleure une touche du clavier.

POUR JOUER :
On lance le programme en tapant RUN.
Le dessin sera ensuite exécuté automatiquement.
Le message FINI (en bas, à gauche) signale la fin du tracé.
Pour exécuter un autre dessin, taper une touche quelconque.

EXTENSIONS POSSIBLES :
On pourra essayer d'utiliser d'autres fonctions pour réaliser des dessins encore plus extraordinaires (par exemple, des dessins simulant une troisième dimension).

MATHIEU

```
10 CLS
20 H=2.5
30 K=5
40 A=3*RND(1)
50 B=3*RND(1)
60 FOR I=0 TO 6.28 STEP .52
70 R=H*SIN(3*I)
80 X=12+INT(K*R*COS(A*I))
90 Y=17+INT(K*R*SIN(B*I))
100 IF (X<1) OR (X>22) OR (Y<2) OR (Y>
32) THEN 120
110 CURSOR Y,X:PRINT "*"
120 NEXT I
130 CURSOR 0,20:PRINT "FINI"
140 K$=INKEY$
150 IF K$="" THEN 140
160 GOTO 10
```

Machinamot

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu de hasard (sur des mots).
Difficulté du jeu : aucune.

LE JEU :
Ce très court programme permet au YENO de faire la démonstration de ses capacités à manipuler des chaînes de caractères : on part simplement d'un préfixe et d'un suffixe d'origine grecque, que l'on va assembler au hasard ; on tombera bien sûr, de temps à autre, sur des mots absurdes ou même légèrement équivoques.
Un nouveau mot est confectionné à chaque essai, mais il est possible d'obtenir plusieurs fois le même mot.

LE PROGRAMME :
La ligne 2Ø permet d'entrer la liste des préfixes dans une variable A$ (il sera bien sûr possible d'ajouter d'autres préfixes, en ne dépassant pas 5 lettres et en ajoutant, si nécessaire, des blancs).
La ligne 3Ø contiendra, de la même façon, la liste des suffixes qui sont placés dans une autre variable alphanumérique B$.
La ligne 4Ø effectue le choix d'un suffixe au hasard.
La ligne 5Ø effectue le choix d'un suffixe, au hasard.
Le mot obtenu est affiché par la ligne 6Ø, et la ligne 8Ø permettra de recommencer un tirage.

POUR JOUER :
Lancer l'exécution du programme par RUN.
Un premier mot sera immédiatement affiché.
Pour obtenir un nouveau mot, il convient d'appuyer sur une touche quelconque (sauf BREAK) ; un nouveau mot sera alors affiché à la suite du précédent.
Taper BREAK pour sortir du programme.

EXTENSIONS POSSIBLES :
On pourra très facilement augmenter la liste des mots disponibles (sur YENO, il sera préférable d'utiliser des instructions DATA qui permettront de ne pas avoir de contraintes sur la longueur des mots).
On pourra aussi utiliser un procédé analogue pour faire confectionner des phrases à l'ordinateur, lui faire inventer des poèmes, etc. On pourra aussi transformer le programme initial en un jeu de société qui consisterait, par exemple, à prendre des paris sur le prochain mot qui va apparaître. Et confier, éventuellement, la gestion des paris au YENO.

MACHINAMOT

```
10 CLS
20 A$="ANDROCOSMOMICROHELIOPEDO PORNOP
ALEOMACROHYDROSTENOPATHOOLEO MYTHOHOME
OOCTO "
30 B$="CRATEPHILEPHOBETROPEGYNE MANE N
AUTEPHAGEFUGE PATHECARDEDERMESCOPEVORE
 PHONE"
40 I=5*INT((LEN(A$)/5)*RND(1))+1
50 J=5*INT((LEN(B$)/5)*RND(1))+1
60 PRINT TAB(5);MID$(A$,I,5);MID$(B$,J
,5)
70 K$=INKEY$
80 IF K$="" THEN 70
90 GOTO 40
100 RUN
```

Compose

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : composition musicale.
Difficulté du jeu : aucune, a priori.

LE JEU :
Il s'agit, en fait, d'un programme capable de composer, de façon très élémentaire, de la musique, et ceci d'une façon complètement aléatoire. L'ordinateur va, dans un premier temps, dessiner une portée, puis il affiche une succession de notes (allant du DO grave au SI aigu) et donne leur nom. Une nouvelle série de notes sera ensuite affichée, et ainsi de suite.

LE PROGRAMME :
La boucle I (2Ø-4Ø) dessine la portée. La variable N$, en ligne 5Ø, contient les noms des notes.
La deuxième boucle I (7Ø-13Ø) permet de faire le choix d'une note de façon aléatoire, à la ligne 8Ø ; son nom est affiché en ligne 9Ø et la note est dessinée par les lignes 1ØØ à 12Ø.
La ligne 15Ø permet de recommencer.

POUR JOUER :
Lancer l'exécution du programme en tapant RUN.
La portée sera alors rapidement affichée, puis les notes sont représentées, avec leur nom affiché en bas de l'écran.
Une pression sur une touche quelconque provoquera l'affichage d'une nouvelle série de notes, et une pression sur la touche BREAK provoque la fin de la partie.

EXTENSIONS POSSIBLES :
On pourra affiner les règles de composition, afficher davantage de notes, introduire des dièses et des bémols, améliorer le graphisme, faire jouer la mélodie affichée.

COMPOSE

```
10 CLS
20 FOR I=1 TO 5
30 CURSOR 3,5+2*I:FOR J= 1 TO 30:PRINT
 CHR$(223);:NEXT J
40 NEXT I
50 N$="SI LA SOLFA MI RE DO "
60 PRINT "";
70 FOR I=1 TO 9
80 X=INT(14*RND(1)+1)
90 CURSOR 0,X+3:PRINT MID$(N$,3*(X+7*(
X>7))-2,3);
100 CURSOR 3*I+2,X+1:PRINT CHR$(227)
110 CURSOR 3*I+2,X+2:PRINT CHR$(227)
120 CURSOR 3*I+2,X+3:PRINT CHR$(236)
130 NEXT I
140 K$=INKEY$
150 IF K$="" THEN 140
160 GOTO 10
```

Nicomaque

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de réflexion (calcul mental).
Difficulté du jeu : moyenne.

LE JEU :
Il utilise une propriété un peu particulière des entiers de 1 à 1ØØ qui peuvent être identifiés sans ambiguïté par leurs restes dans les divisions par 3, 5 et 7.
Le programme, proposé ici, est un programme codeur qui affiche simplement les restes d'un entier donné N dans les divisions indiquées.
Le jeu consiste à essayer de découvrir l'entier (mais, dans cette version, vous n'avez droit qu'à un seul essai).

LE PROGRAMME :
La ligne 5Ø permet de confectionner l'entier X.
La boucle I (6Ø-12Ø) calcule les trois restes successifs, et la boucle emboîtée J (9Ø-11Ø) permet de les afficher sous une forme assez visuelle, aux bons emplacements.
La ligne 13Ø est là pour indiquer de quelle division il s'agit.
Le nombre proposé par le joueur est entré à la ligne 14Ø . Il est testé dans une variable logique R$ qui permettra ensuite, l'affichage de fin de partie (BRAVO ou PERDU).
La ligne 15Ø réalise l'affichage de fin.
Les lignes 16Ø et 17Ø s'occupent de gérer la prochaine partie, selon la réponse du joueur.

POUR JOUER :
On lance le programme en tapant RUN.
Les trois restes seront alors affichés sur la même ligne, symbolisés par des points (l'absence de points signifiant : reste Ø).
A la question VOTRE ESSAI, on répondra en donnant un entier compris entre Ø et 1ØØ, suivi de CR.
On obtiendra, en réponse, un affichage PERDU ou BRAVO, puis la question AUTRE PARTIE (O/N) ?
Pour faire un nouvel essai, on répondra O, suivi de CR.

EXTENSIONS POSSIBLES :
On peut envisager de réaliser le programme décodeur (on donne les restes et le YENO devra vous donner N) et chercher si la propriété peut être étendue (en effectuant davantage de divisions, par exemple).

NICOMAQUE

```
10 DIM R$(1)
20 R$(0)="PERDU"
30 R$(1)="BRAVO"
40 CLS
50 X=INT(100*RND(1)+1)
60 FOR I=1 TO 3
70 A=X-(I+I+1)*INT(X/(I+I+1))
80 IFA=0 THEN 120
90 FOR J=1 TO A
100 CURSOR 2+8*I+J,10:PRINT CHR$(249)
110 NEXT J
120 NEXT I
130 PRINT :PRINT "RESTE      PAR 3   P
AR 5   PAR 7  ":PRINT :PRINT :PRINT :P
RINT :PRINT
140 INPUT"VOTRE ESSAI ";N
150 PRINT :PRINT R$(-(N=X)):PRINT
160 INPUT "VOTRE PARTIE (O/N) ? ";T$
170 IF T$="O" THEN 40
```

Permutations

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion.
Difficulté du jeu : moyenne.

LE JEU :
Le YENO vous demande d'abord le nombre de lettres que vous désirez permuter (pas plus de 12). A partir de là, il va vous afficher le nombre correspondant de lettres prises dans le début de l'alphabet, d'abord dans le désordre (DEBUT) puis dans l'ordre (FIN).
Vous devez réussir à remettre dans l'ordre les lettres données au début, en utilisant des permutations sur les n premières lettres à partir de la gauche, avec le moins d'essais possible.

LE PROGRAMME :
La ligne 3Ø permet de choisir le nombre de lettres, et la ligne 5Ø contrôle que l'on n'en demande pas plus de 12.
La ligne 7Ø réalise le brouillage, en tirant au hasard la nouvelle position de chaque lettre. Les lettres sont ensuite ajoutées au fur et à mesure de leur tirage, dans une variable B$ (ligne 9Ø), supprimées dans A$ et replacées en C$, dans le bon ordre (lignes 11Ø-12Ø).
La ligne 15Ø contrôle le nombre de lettres de B$.
Les affichages sont effectués aux lignes 17Ø et 18Ø.
Le retournement est fait, lettre par lettre, dans la boucle I (22Ø-24Ø) ; le résultat est placé dans B$ en ligne 25Ø, affiché par la ligne 26Ø et comparé au résultat final en ligne 27Ø.
Le nombre d'essais sera affiché en fin de partie, grâce à la ligne 28Ø.

POUR JOUER :
Après avoir tapé RUN, on obtient l'affichage de la question NBRE DE LETTRES (DE 1 A 12) ; on répond en frappant N, suivi de CR. L'affichage suivant présente l'ordre initial (DEBUT) et l'ordre à obtenir (FIN).
On joue en indiquant le nombre de lettres à permuter à partir de la gauche, par exemple : sur un affichage initial de GEABDFC, si l'on tape 3 et CR, on obtiendra AEGBDFC.
On poursuivra jusqu'à l'obtention de l'ordre désiré.
Le score sera affiché en fin de partie ; ainsi qu'une question AUTRE PARTIE (O ou N) ; répondre O si l'on désire faire une autre partie.

EXTENSIONS POSSIBLES :
On pourra utiliser le même type de jeu pour, par exemple, reconstituer un dessin caché dont les diverses parties seraient repérées par des lettres. On peut aussi réaliser des permutations plus complexes (par exemple le fait de permuter deux lettres permute les deux voisines de la lettre la plus à gauche, etc.).

PERMUTATIONS

```
10 CLS
20 A$="ABCDEFGHIJKL"
30 INPUT "NOMBRE DE LETTRES (1-12) :";
N
40 C$="            "
50 IF N>12 THEN 10
60 B$=""
70 X=INT(RND(1)*N+1)
80 IF MID$(A$,X,1)=" " THEN 70
90 B$=MID$(A$,X,1)+B$
100 Y=INT(RND(1)*N+1)
110 IF MID$(C$,Y,1)<>" " THEN 100
120 C$=LEFT$(C$,Y-1)+MID$(A$,X,1)+RIGH
T$(C$,12-Y-1)
130 A$=LEFT$(A$,X-1)+RIGHT$(A$,12-X)
140 S=1:IF LEFT$(B$,N)=LEFT$(C$,N) THE
N 40
150 IF LEN(B$)<>N THEN 80
160 C$=LEFT$(C$,LEN(B$))
170 PRINT :PRINT "DEBUT    : ";B$:PRINT
 "FIN      : ";C$
180 PRINT :PRINT :PRINT "PERMUTATION "
+STR$(S)+" :";
190 INPUT L
200 S=S+1
210 A$=""
220 FOR I=1 TO L
230 A$=A$+MID$(B$,L-I+1,1)
240 NEXT I
250 B$=A$+MID$(B$,L+1,12)
260 PRINT "RESULTAT: ";B$:PRINT
270 IF B$<>C$ THEN 180
280 PRINT :PRINT :PRINT :PRINT "TROUVE
 EN ";S-1;" ESSAIS"
290 INPUT "UNE AUTRE PARTIE (O/N) :";A
$
300 IF A$="O" THEN 10
```

Tour de cartes

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu de prestidigitation.
Difficulté du jeu : facile.

LE JEU :
C'est un tour assez classique mais, ici, les cartes sont manipulées sous vos yeux par le YENO qui tient le rôle du prestidigitateur ; 21 cartes, prises au hasard dans un paquet de 52, sont affichées, et le joueur devra retenir mentalement l'une des cartes présentées. Puis, il devra indiquer au YENO dans quelle colonne (1re, 2e ou 3e à partir de la gauche) se trouve la carte choisie ; les cartes sont alors redistribuées et le joueur indique de nouveau la colonne correspondant à sa carte ; ceci est fait encore une fois et, au 4e affichage, le YENO est en mesure de deviner la carte choisie.

LE PROGRAMME :
La boucle I (5Ø-11Ø) permet de choisir, au hasard, 21 cartes sur 52, et la boucle emboîtée J (7Ø-9Ø) permet de s'assurer que l'on ne réalise pas deux fois le même tirage. La ligne 14Ø contient les couleurs des cartes ainsi que les noms des cartes figures (VA, DA, RO, AS), dans une variable alphanumérique B$.
La boucle I (15Ø-27Ø) réalise l'affichage du jeu.
La ligne 16Ø décode la couleur de la carte.
La ligne 17Ø choisit la position d'affichage.
La ligne 18Ø décode la hauteur.
L'affichage proprement dit est effectué en ligne 26Ø.
La ligne 3ØØ permet au joueur d'entrer le numéro de la colonne où se trouve sa carte. La boucle J (35Ø-37Ø) réalise le brouillage, et les boucles emboîtées J (43Ø-47Ø) et I (44Ø-46Ø) fabriquent la nouvelle variable A$ à utiliser pour le prochain affichage.

POUR JOUER :
On lance le programme par RUN.
On obtient un premier affichage de 21 cartes réparties sur 3 colonnes décalées.
On indique alors le numéro (1, 2 ou 3) de la colonne où se trouve la carte choisie, suivi de CR.
On obtient un 2e affichage et l'on fait de même ; puis un 3e affichage et l'on fait de même.
Au 4e affichage, la carte choisie est la dernière nommée par le YENO.

EXTENSIONS POSSIBLES :

On peut envisager une amélioration de l'affichage en écrivant les noms des cartes en entier, ou même, en dessinant de véritables cartes, en couleur. On pourra aussi essayer de programmer d'autres tours, utilisant le même principe de base, mais avec davantage de cartes, etc.

```
           TOUR DE CARTES

10 CLS
20 A$="ABCDEFGHIJKL"
30 INPUT "NOMBRE DE LETTRES (1-12) :";
N
40 C$="            "
50 IF N>12 THEN 10
60 B$=""
70 X=INT(RND(1)*N+1)
80 IF MID$(A$,X,1)=" " THEN 70
90 B$=MID$(A$,X,1)+B$
100 Y=INT(RND(1)*N+1)
110 IF MID$(C$,Y,1)<>" " THEN 100
120 C$=LEFT$(C$,Y-1)+MID$(A$,X,1)+RIGH
T$(C$,12-Y-1)
130 A$=LEFT$(A$,X-1)+" "+RIGHT$(A$,12-
X)
140 S=1
150 IF LEN(B$)<>N THEN 80
160 C$=LEFT$(C$,LEN(B$))
170 PRINT :PRINT "DEBUT   : ";B$:PRINT
 "FIN     : ";C$
180 PRINT :PRINT :PRINT "PERMUTATION "
+STR$(S)+" :";
190 INPUT L
200 S=S+1
210 A$=""
220 FOR I=1 TO L
230 A$=A$+MID$(B$,L-I+1,1)
240 NEXT I
250 B$=A$+MID$(B$,L+1,12)
260 PRINT "RESULTAT : ";B$:PRINT
270 IF B$<>C$ THEN 180
280 PRINT :PRINT :PRINT :PRINT "TROUVE
 EN ";S-1;" ESSAIS"
290 INPUT "UNE AUTRE PARTIE (O/N) :";A
$
300 IF A$="O" THEN 10
```

Parcours du cavalier

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion.
Difficulté du jeu : difficile.

LE JEU :
Il se présente, ici, comme un jeu de hasard total, mais provient, en réalité, d'un problème assez difficile à résoudre : comment déplacer un cavalier sur un échiquier de façon qu'il passe une fois, et une seule, sur chacune des 64 cases du damier.
Le cavalier se déplace suivant les règles communément admises aux échecs, et sa position de départ est définie par le joueur ; par contre, ses déplacements sont effectués au hasard. Arrivera-t-il à parcourir les 64 cases ?

LE PROGRAMME :
Les lignes 3Ø et 4Ø définissent les divers déplacements possibles, dans deux variables alphanumériques A$ et B$; les deux boucles emboîtées I (5Ø-9Ø) et J (6Ø-8Ø) dessinent le terrain de jeu.
On entre les coordonnées de la case de départ (ligne, puis colonne) ligne 1ØØ.
La ligne 17Ø permet le tirage au hasard d'un déplacement du cavalier.
La boucle Z (18Ø-24Ø) cherche si ce déplacement est possible ; si ce n'est pas le cas, on essaie tous les autres (toujours dans le même ordre).
Les lignes 19Ø et 2ØØ effectuent le déplacement.
La ligne 21Ø contrôle que l'on ne sorte pas du terrain de jeu.
la ligne 23Ø envoie en ligne 27Ø si le coup est possible.
La ligne 26Ø gère la fin de partie (cavalier bloqué) et affiche le score.
Les lignes 28Ø, 29Ø et 3ØØ réalisent le coup et incrémentent le score.

POUR JOUER :
On lance le programme par l'ordre RUN. Le terrain de jeu s'affiche ; on choisit alors la position de départ en tapant d'abord un numéro de ligne (entre 1 et 8), puis un numéro de colonne, suivi de CR. Le jeu commence immédiatement et il se déroule de façon entièrement automatique jusqu'à ce que le cavalier soit bloqué. On peut, bien sûr, arrêter à tout moment par BREAK.

EXTENSIONS POSSIBLES :
On peut imaginer de nombreuses variantes sur ce thème, ne serait-ce qu'en commençant par faire varier la taille des échiquiers, on peut aussi améliorer considérablement le graphisme, utiliser la couleur. Mais, le plus intéressant est d'arriver à définir un algorithme donnant la ou les solutions du problème initial...

PARCOURS DU CAVALIER

```
10 CLS
20 S=1
30 A$="56653223"
40 B$="653223"+A$+"56"
50 FOR I=S TO LEN(A$)
60 FOR J=S TO LEN(A$)
70 CURSOR 3*J+3,2*I:PRINT "."
80 NEXT J
90 NEXT I
100 CURSOR 0,20:INPUT I,J
110 A$=A$+A$
120 IF S>9 THEN 150
130 CURSOR 3*J+3,2*I:PRINT CHR$(48+S)
140 GOTO 170
150 CURSOR 3*J+2,2*I:PRINT CHR$(48+INT
(S/10))
160 CURSOR 3*J+3,2*I:PRINT CHR$(48+S-I
NT(S/10)*10)
170 Z=INT(8*RND(1)+1)
180 FOR Z=Z TO Z+8
190 A=I+VAL(MID$(A$,Z,1))-4
200 B=J+VAL(MID$(B$,Z,1))-4
210 IF (A<1) OR (B<1) OR (A>8) OR (B>8
) THEN 240
220 C=VPEEK(40*2*A+3*B+3+15362)
230 IF C=46 THEN 270
240 NEXT Z
250 CURSOR 0,20:PRINT ,"MAX :" ;S
260 END
270 I=A
280 J=B
290 S=S+1
300 GOTO 120
```

Mur de briques

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexe (pour 1 joueur).
Difficulté du jeu : difficile.

LE JEU :
S'il est un jeu que l'on peut présenter comme un grand classique des jeux d'arcades pour micro-ordinateur, c'est bien le mur de briques (en anglais BREAKOUT).
Il est offert, ici, dans une version simplifiée, mais offre néanmoins de bonnes performances : affichage du score en fin de partie, rebonds aléatoires, etc.
La règle en est très élémentaire : vous disposez d'une raquette qui peut être déplacée horizontalement pour renvoyer une balle (matérialisée par un zéro) qui va démolir les briques dessinées en haut de l'écran.
Votre problème est d'en démolir le plus grand nombre possible, sans rater la balle, ce qui provoquerait la fin de partie et l'affichage du score.

LE PROGRAMME :
La boucle I (5Ø-7Ø) et les lignes 8Ø et 9Ø dessinent le terrain de jeu et les briques.
La ligne 12Ø dessine la raquette dont la position est contrôlée par la ligne 14Ø.
Les lignes 23Ø, 24Ø et 25Ø regardent si l'on a touché le mur, les briques ou la raquette.
Les rebonds seront gérés en conséquence avec des éventuels aléas, par les lignes 29Ø à 35Ø.
La ligne 17Ø affiche le score en fin de partie.

POUR JOUER :
Le programme sera lancé par l'instruction RUN, ce qui provoque l'affichage du terrain de jeu puis, très rapidement, l'envoi de la balle. Le joueur devra alors essayer de déplacer sa raquette à l'aide des touches de déplacement horizontal.
Si, à un moment quelconque, sa balle ne touche pas la raquette, le score sera affiché et la partie s'arrête.
On peut aussi terminer à tout moment par BREAK.

EXTENSIONS POSSIBLES :
On peut envisager d'agrandir le terrain de jeu, d'augmenter le nombre de briques, de prévoir plusieurs balles, de dessiner chaque rangée de briques d'une couleur différente, d'introduire des effets sonores.
On peut aussi, en s'inspirant des jeux d'arcades, introduire de petits ouvriers qui viendront reconstruire votre mur de briques, au fur et à mesure que vous le démolissez.

MUR DE BRIQUES

```
10 CLS
20 S=0
30 Y=14
40 J=14
50 FOR I=0 TO 7
60 CURSOR 10,I:PRINT "############"
70 NEXT I
80 FOR L=0 TO 21:CURSOR 9,L:PRINT ":";
:NEXT L
90 FOR L=0 TO 21 :CURSOR 22,L:PRINT ":
":NEXT L
100 H=-1
110 K=H
120 CURSOR Y,21:PRINT " ";CHR$(157);CH
R$(157);CHR$(157);" "
130 K$=INKEY$
140 Y=Y+(K$=CHR$(28))*(Y<18)-(K$=CHR$(
29))*(Y>9)
150 CURSOR I,J:PRINT " "
160 IF I<21 THEN 200
170 CURSOR 25,10:PRINT "SCORE :";S
180 BEEP
190 GOTO 190
200 I=I+H
210 J=J+K
220 C=VPEEK(I*40+J+15362)
230 IF C=32 THEN 270
240 IF C=58 THEN 310
250 IF C=35  THEN 330
260 GOTO 290
270 CURSOR I,J:PRINT CHR$(249)
280 GOTO 120
290 H=-H
300 K=RND(1)-RND(1)
310 K=-K
320 GOTO 200
330 S=S+1
340 K=-(K+(2*(RND(1)<.5)-1)*RND(1)/3)
350 H=-H
360 GOTO 120
```

Niveau 4

Nom du programme	*Thèmes et particularités*
Tirage de dés	Suffisamment souple pour être utilisé partout.
Jeu de loto	Gagnez sans problème, au prochain tirage.
La bombe	Saurez-vous la découvrir à temps ? (1 joueur).
Ardoise magique	Mettez au point vos créations graphiques, sans papier ni crayon.
Affichage publicitaire	Employez les possibilités de traitement de chaîne du YENO.
Evasion	Il vous faudra beaucoup de chance pour réussir.
Chapeaux	Soyez très observateur, et vous réussirez (2 joueurs).
Combat aérien	Pas si facile de descendre un appareil qui bouge tout le temps.
Dessin de dés	Améliorez vos jeux de dés en incorporant ce programme.
Le jeu du missile	Une mission bien périlleuse ! (un seul joueur).
Le singe	Un jeu de lettres pour 2 joueurs, où il faut de bons réflexes.
Le jeu du 15	Un jeu de réflexion utilisant des chiffres (2 joueurs).
Adresse	Jeu de réflexe pour un joueur, très visuel.
Sous-marin	Un classique des jeux vidéos (1 joueur).
Zombies	Un peu farfelu, mais très visuel (1 joueur).
Diago	Etes-vous vraiment sûr de vos réflexes ? (1 joueur).
Topologie	Une direction à explorer, pour d'autres jeux (1 ou 2 joueurs).
Chasse au sous-marin	Il vous faudra pas mal de réflexion et beaucoup de chance.
Le lézard d'Alice	Beaucoup de chance, mais très spectaculaire (1 joueur).
Calculatrice	Un programme de simulation qui pourra facilement être amélioré.
Smokey	Fumez sans risque, avec ce programme original pour 1 joueur.

Course de voitures	Un jeu de réflexes, pas trop difficile, pour un joueur.
Echecs	Un programme très simplifié, pour deux joueurs.
Danger spatial	Serez-vous assez adroit pour éviter toutes les météorites ?
Tic tac toe	Un classique, mais pas sans intérêt (1 joueur).
Gouttes	Plutôt un gag visuel et sonore. A exploiter.
Obstacles	Beaucoup d'obstacles à éviter dans ce jeu d'adresse pour un joueur.
Captures dans l'espace	Votre mission serait facile, sans ces damnées météorites !
Plusieurs en un	Un terrain de jeu, et des pions, pour deux joueurs. Choisissez !

Les techniques d'animation élémentaire

SAISIE AU VOL D'UN CARACTERE FRAPPE AU CLAVIER : INKEY$

Le principe d'un jeu graphique, en animation, est que le jeu doit se dérouler tout seul, d'une certaine manière, pendant un certain temps, tant qu'aucune intervention n'a été faite par le joueur.

Il faut donc disposer de la possibilité d'observer s'il y a eu ou non intervention du joueur, sans pour cela interrompre le déroulement du programme.

Dans le Basic YENO, cette propriété est offerte par la fonction INKEY$ qui effectue une scrutation de toutes les touches du clavier et enregistre le code de la touche frappée.

On pourra utiliser INKEY$ de deux façons :

- soit dans une boucle (réalisation d'un mouvement), on saisit une variable au moyen de INKEY$ et on la teste pour sortir éventuellement de la boucle ;

- soit, au contraire, tant que la variable saisie est nulle (on n'a rien frappé), on attend et l'on sort dès que l'on a frappé quelque chose.

Rappel : La syntaxe de l'instruction INKEY$ est : K$ = INKEY$, où K$ est la valeur de la variable saisie (chaîne nulle – K$ = " " – si aucune touche n'a été enfoncée).

Tirage de dés

REFERENCES :
Difficulté de programmation : aucune.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : aucune.

LE JEU :
Le programme proposé ne réalise pas un jeu en lui-même, mais pourra être utilisé en sous-programme pour de nombreux jeux.
Il offre simplement la possibilité de lancer de 1 à 9 dés, par simple pression d'une touche. Les dés sont ensuite affichés avec un graphisme très simplifié.

LE PROGRAMME :
La ligne 4Ø saisit le caractère frappé au clavier et le transforme en nombre. Une boucle J (7Ø-9Ø) est utilisée pour dessiner un fond gris, et une autre boucle I (1ØØ-16Ø) sert au tirage (tirage en ligne 14Ø et impression en ligne 15Ø).

POUR JOUER :
On lance le programme par l'ordre RUN et on obtient l'affichage NOMBRE DE DES A LANCER auquel on répond en frappant un chiffre.
Le résultat est alors automatiquement affiché.

EXTENSIONS POSSIBLES :
On pourra, bien sûr, utiliser le canevas fourni par ce programme pour la réalisation de tout jeu se servant de dés (421, YAM'S, CRAPS, etc.). On peut prévoir, en fin de programme, une instruction qui relance automatiquement le tirage.

TIRAGES DE DES

```
10 CLS
20 PRINT "NOMBRE DE DES A LANCER :"
30 BEEP
40 K$=INKEY$
50 IF K$="" THEN 40
60 N=VAL(K$)
70 FORJ=1 TO 3
80 CURSOR 2,8+J:PRINT CHR$(144);CHR$(1
44);CHR$(144)
90 NEXT J
100 FOR I=1 TO N
110 FOR J=1 TO 3
120 CURSOR 3*I,8+J:PRINT CHR$(144);CHR
$(144);CHR$(144)
130 NEXT J
140 D(I)=INT(6*RND(1)+1)
150 CURSOR 3*I,10:PRINT D(I)
160 NEXT I
170 CURSOR 0,20
```

Jeu de loto

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : hasard (plusieurs joueurs).
Difficulté du jeu : facile.

LE JEU :
il s'agit de réaliser un tirage complet de loto, c'est-à-dire d'obtenir 6 numéros, compris entre 1 et 49, plus un numéro complémentaire, tous différents, qui pourraient être, par exemple, utilisés pour remplir votre prochaine grille de loto.

LE PROGRAMME :
On utilisera un tableau N(7), dimensionné en ligne 1Ø, pour enregistrer les résultats.
Le tirage des nombres est effectué dans une boucle I (3Ø-9Ø), une deuxième boucle J (6Ø-8Ø) étant utilisée pour contrôler qu'il n'y a pas deux tirages identiques.
Le tirage (ou le retirage) est effectué en ligne 4Ø.
La boucle I (11Ø-13Ø) affiche le résultat et la ligne 14Ø affiche le numéro complémentaire.
Les lignes 15Ø à 17Ø s'occupent du cas où l'on voudrait refaire une partie.

POUR JOUER :
Il suffit de taper RUN pour lancer le programme et obtenir l'affichage d'un premier loto.
La question VOULEZ-VOUS UN AUTRE TIRAGE (O ou N) ? s'affiche en bas de l'écran ; on répondra O si l'on désire continuer.

EXTENSIONS POSSIBLES :
On peut imaginer un programme qui pourrait tenir compte, soit de ses propres résultats, soit de résultats entrés par un joueur, pour améliorer le tirage, en tenant compte des probabilités de sortir un numéro donné.
On peut aussi envisager un programme qui réaliserait des cartons pour le jeu de loto plus usuel, tel qu'il est encore couramment pratiqué dans le midi de la France.

JEU DE LOTO

```
10 DIM N(7)
20 CLS
30 FOR I=1 TO 7
40 N(I)=INT(49*RND(1)+1)
50 IF I=1 THEN 90
60 FOR J=1 TO I-1
70 IF N(I)=N(J) THEN 40
80 NEXT J
90 NEXT I
100 PRINT TAB(5);"TIRAGE DU LOTO":PRIN
T :PRINT :PRINT :PRINT
110 FOR I=1 TO 6
120 PRINT TAB(4*I);N(I);
130 NEXT I
140 PRINT :PRINT :PRINT "NUMERO COMPLE
MENTAIRE :";N(7):PRINT :PRINT "VOULEZ
VOUS UN AUTRE TIRAGE ? (O/N)"
150 K$=INKEY$
160 IF K$="" THEN 150
170 IF K$="O" THEN 20
```

La bombe

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : solitaire.
Difficulté du jeu : moyenne.

LE JEU :
Vous êtes sensé trouver, à l'aide d'un détecteur spécialisé, un colis radioactif piégé disposé par un voyageur mal intentionné, dans une chambre d'hôtel. Seul problème, mais de taille : l'hôtel comporte un million de chambres (1ØØ*1ØØ*1ØØ).
Votre détecteur affiche, en permanence, un signal qui vous renseigne sur la proximité de la bombe, et le temps qui vous reste est affiché en permanence, décompté de 1Ø en 1Ø.
Pourrez-vous découvrir la bombe avant l'explosion ?

LE PROGRAMME :
Une variable numérique R est dimensionnée à 3 en ligne 3Ø pour trois coordonnées définissant la position du joueur.
La position de la chambre contenant le colis piégé est définie par un tirage aléatoire de 3 variables A, B, C, aux lignes 4Ø, 5Ø et 6Ø.
Les lignes 9Ø à 13Ø dessinent le détecteur.
Le signal de proximité est calculé en ligne 8Ø et la ligne 16Ø permet l'affichage des coordonnées de la chambre à découvrir, dans le cas où l'on a perdu.
La ligne 1ØØ permet l'affichage, à la fois du signal et du temps restant à courir.
La ligne 18Ø réalise le décompte du temps.
La ligne 19Ø contrôle si l'on a trouvé ou non.
La boucle I (25Ø-29Ø) permet d'entrer les nouvelles coordonnées, en utilisant le sous-programme placé aux lignes 31Ø à 4ØØ.

POUR JOUER :
Après le lancement du programme par RUN, le signal et le temps (en principe 2ØØ) sont affichés.
Le joueur est alors en position x=Ø, y=Ø, z=Ø, et on devra introduire successivement les trois coordonnées x, y, z de la chambre à explorer.
On fera, par exemple, 13, puis 45, puis 83, puis CR, etc.
Le nouveau signal est alors affiché, et le temps est décrémenté de 1Ø.
On continuera ainsi, soit jusqu'au bon résultat (arrêt du jeu et fin de programme), soit jusqu'à l'apparition de 3 nombres (x,y,z) sans sortie du programme, ce qui signifie alors que l'on a perdu.

EXTENSIONS POSSIBLES :
Il sera sans doute indispensable d'améliorer le graphisme, d'ajouter une explosion (sonore et visuelle) pour symboliser l'éclatement de la bombe, d'afficher symboliquement la position dans l'hôtel, etc.

EXTENSIONS POSSIBLES :

Il sera sans doute indispensable d'améliorer le graphisme, d'ajouter une explosion (sonore et visuelle) pour symboliser l'éxplosion de la bombe, d'afficher symboliquement la position dans l'hôtel, etc.

LA BOMBE

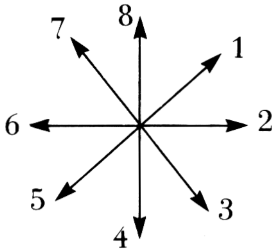
```
10 L=100
20 T=L+L
30 DIM R(3)
40 A=INT(L*RND(1))
50 B=INT(L*RND(1))
60 C=INT(L*RND(1))
70 CLS
80 S=ABS((A/L+B+C*L)-(R(1)/L+R(2)+R(3)
*L))
90 CURSOR 5,22:FOR J=1 TO 24:PRINT CHR
$(229);:NEXT J:PRINT
100 PRINT TAB(11);INT(S);TAB(20);T
110 CURSOR 5,22:FOR J=1 TO 5:PRINT CHR
$(229);:NEXT J
120 CURSOR 17,22:PRINT CHR$(229);CHR$(
229);CHR$(229)
130 CURSOR 24,22:FOR J=1 TO 5:PRINT CH
R$(229);:NEXT J
140 IF T<>0 THEN 180
150 CLS
160 PRINT :PRINT :PRINT "PERDU :";A;B;
C;
170 END
180 T=T-10
190 IF S<>0 THEN 230
200 CLS
210 PRINT :PRINT :PRINT "BRAVO"
220 END
230 PRINT :PRINT :PRINT :PRINT :PRINT
,">X=?":PRINT :PRINT ,">Y=?":PRINT :PR
INT ,">Z=?":PRINT :PRINT :PRINT :PRINT

240 CURSOR5,11:FOR J=1 TO 24:PRINT CHR
$(229);:NEXT J
250 FOR I=1 TO 3
260 R$=""
270 GOSUB 310
280 R(I)=VAL(R$)
290 NEXT I
300 GOTO 70
310 CC=22
320 BEEP
330 K$=INKEY$
340 IF K$="" THEN 330
350 IF K$=CHR$(13) THEN 400
360 R$=R$+K$
370 CURSOR CC,12+2*I:PRINT K$
380 CC=CC+1
390 GOTO 330
400 RETURN
```

Ardoise magique

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : dessin.
Difficulté du jeu : facile.

LE JEU :
Comme le titre peut sembler l'indiquer, il s'agit, tout simplement d'utiliser l'écran du téléviseur comme une ardoise, en manipulant un point sur l'écran suivant 8 directions, à l'aide des touches de 1 à 8, suivant la convention :



On dispose aussi de la possibilité de déplacer le point sur l'écran sans laisser de trace en appuyant sur la touche Ø (ce qui permettra aussi l'effacement partiel) et l'on pourra réinscrire en tapant 9.
Pour effacer complétement un dessin, on devra frapper la touche P.

LE PROGRAMME :
La ligne 5Ø réalise l'affichage d'un point sur l'écran.
Les lignes 6Ø et 7Ø saisissent au vol la touche frappée (de Ø à 9, ainsi que P).
La ligne 16Ø appelle l'un des sous-programmes effectuant le déplacement.
Les lignes 17Ø et 18Ø contrôlent que l'on ne sorte pas de l'écran.
La ligne 8Ø permet de tout effacer si l'on appuie sur la touche P.

POUR JOUER :
On lance le programme en tapant RUN , puis on pourra déplacer le point qui apparaît en bas à gauche de l'écran, en utilisant les 8 touches de déplacement.
La touche Ø permet de déplacer le point sans laisser de trace (il y aura effacement si l'on repasse sur un ancien trait).
La touche 9 permet d'inscrire le déplacement.
La touche P efface tout, et l'on recommance.
Pour sortir du programme, il faut taper BREAK.

EXTENSIONS POSSIBLES :
On pourra envisager des possibilités particulières pour tracer des droites, des cercles, dessiner avec plusieurs couleurs, améliorer la finesse.
On pourra ajouter, si besoin est, un programme qui sauve le dessin obtenu pour une utilisation ultérieure.

ARDOISE MAGIQUE

```
10 CLS
20 X=4
30 Y=20
40 S=0
50 CURSOR X,Y:PRINT CHR$(229)
60 K$=INKEY$
70 IF K$="" THEN 60
80 IF K$="P" THEN 10
90 IF (ASC(K$)<48) OR (ASC(K$)>57) THE
N 60
100 IF K$="0" THEN 230
110 IF K$="9" THEN 40
120 R=ASC(K$)-48
130 F=-2*(R<5)-1
140 A=X
150 B=Y
160 ON R+2*(F-1) GOSUB 250,260,280,290
,310
170 C=(X>=0)*(X<=36)*(Y>=0)*(Y<=22)
180 IF C<>0 THEN 200
190 GOSUB 310
200 IF S=0 THEN 220
210 CURSOR A,B:PRINT " "
220 GOTO 50
230 S=1
240 GOTO 50
250 Y=Y-F
260 X=X+F
270 RETURN
280 GOSUB 260
290 Y=Y+F
300 RETURN
310 X=A
320 Y=B
330 RETURN
```

Affichage publicitaire

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : affichage publicitaire.
Difficulté du jeu : aucune.

LE JEU :
Un texte quelconque devra être introduit en début de programme. Il pourra être relativement long (4 ou 5 lignes) ; on choisira ensuite le sens et la vitesse de défilement, à la suite de quoi le texte va apparaître (une seule ligne à la fois) et commencera à défiler, à la vitesse choisie, et dans le sens choisi.

LE PROGRAMME :
Le texte sera introduit dans une variable P$ à la ligne 2Ø.
L'affichage sera tronqué à 28 caractères au plus, grâce à la ligne 13Ø.
Le défilement vers la gauche est préparé à la ligne 18Ø, et le défilement vers la droite à la ligne 2ØØ.
Le texte sera affiché en ligne 16Ø, et la durée de l'affichage (vitesse de défilement) est choisie par la ligne 21Ø.
La ligne 23Ø permettra de changer, si besoin est, les conditions du défilement.

POUR JOUER :
On lancera le programme par RUN, et à l'affichage de la question VOTRE TEXTE, on introduira le texte à faire défiler, suivi de CR.
On obtient ensuite l'affichage SENS D ou G auquel on répondra en tapant, soit D, soit G, suivi de CR.
Puis à la question VITESSE, on répondra en introduisant un petit nombre pour une vitesse rapide, et un grand nombre pour une vitesse lente (par exemple, de Ø à 4Ø ou plus).
Le texte commencera à défiler dès que l'on aura frappé CR.
Si l'on désire changer les conditions du défilement, il suffit d'appuyer sur la touche O pour retrouver les questions SENS ET VITESSE. On tapera BREAK pour interrompre le programme.

EXTENSIONS POSSIBLES :
On pourra introduire de la couleur, réaliser un affichage ondulant, faire remonter les lignes sur l'écran, etc.

AFFICHAGE PUBLICITAIRE

```
10 CLS
20 INPUT "ENTREZ VOTRE TEXTE :";P$
30 P$=P$+" "
40 INPUT "DEFILEMENT G OU D :";S$
50 IF LEFT$(S$,1)="G"  THEN 80
60 PRINT "DROITE"
70 GOTO 90
80 PRINT "GAUCHE"
90 INPUT "VITESSE DE 0 A 50 :";V
100 V=51-V
110 T=LEN(P$)
120 CLS
130 IF T<=28 THEN 160
140 T=28
150 CURSOR 0,0
160 PRINT MID$(P$,1,T)
170 IF LEFT$(S$,1)="D" THEN 200
180 P$=RIGHT$(P$,LEN(P$)-1)+LEFT$(P$,1
)
190 GOTO 210
200 P$=RIGHT$(P$,1)+LEFT$(P$
)
210 FOR P=1 TO V
220 NEXT P
230 K$=INKEY$
240 IF K$="0" THEN 40
250 GOTO 150
```

Evasion

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : solitaire.
Difficulté du jeu : facile.

LE JEU :
Votre seul problème : réussir à traverser l'écran, sans que le YENO vous ait auparavant touché par l'un des nombreux obus explosifs qu'il va tirer sur vous.
Vous pouvez vous déplacer dans 4 directions (à l'aide des 4 flèches), et vous devez réussir à sortir du terrain par le bord droit, sans, bien sûr, avoir été touché une seule fois.

LE PROGRAMME :
Les lignes 3Ø et 4Ø définissent la position de départ du joueur.
La ligne 5Ø choisit la coordonnée horizontale de la case où YENO va tirer (liée à votre position), et la ligne 6Ø choisit la coordonnée verticale (arbitraire) de cette case.
Le déplacement du joueur est contrôlé par les lignes 12Ø et 13Ø ; sa trajectoire est tracée par la ligne 1ØØ, et sa position actuelle (sur l'écran, le caractère 128) est imprimée par la ligne 15Ø.
La ligne 16Ø regarde s'il est touché ou non.
La ligne 19Ø effectue le contrôle de fin de partie.

POUR JOUER :
On commence l'exécution du programme par RUN.
Un premier tir a lieu, et ce n'est qu'après que le joueur apparaît, représenté par le caractère 128.
Les 4 touches de direction servent à déplacer le joueur.
S'il est touché, on obtient, sur l'écran, le message PERDU et la partie s'arrête.
S'il réussit à sortir par la droite de l'écran sans être touché, on obtient le message BRAVO, et la partie est également terminée.
On peut terminer à tout moment en tapant BREAK.

EXTENSIONS POSSIBLES :
La couleur et un graphisme plus soigné amélioreront considérablement l'attrait du jeu. On pourra aussi rendre le tir des bombes plus aléatoire, introduire des obstacles à contourner, ajouter des contrôles, faire participer plusieurs joueurs, etc.

EVASION

```
10 CLS
20 PATTERN C# 128, "1818087E08182424"
30 X=10
40 Y=10
50 H=2*(Y-1)+6*RND(1)
60 K=19*RND(1)+3
70 FOR J=1 TO 7*RND(1)+3
80 CURSOR ABS(H+4*RND(1)-2),ABS(K+4+RN
D(1)-2):PRINTCHR$(229)
90 NEXT J
100 CURSOR X,Y:PRINT"."
110 K$=INKEY$
120 X=X+(K$=CHR$(29))-(K$=CHR$(28))
130 Y=Y+(K$=CHR$(30))-(K$=CHR$(31))
140 U=VPEEK(40*Y+X+15362)
150 CURSOR X,Y:PRINTCHR$(128)
160 IF U<>229 THEN 190
170 CURSOR 0,0:PRINT"PERDU"
180 GOTO 210
190 IF X<37 THEN 50
200 CURSOR 0,0:PRINT"BRAVO"
210 END
```

Chapeaux

REFERENCES :
Difficulté de programmation : assez facile.
Catégorie de jeu : jeu de réflexe, pour deux joueurs.
Difficulté du jeu : facile.

LE JEU :
L'idée en est très simple : deux personnages se déplacent l'un vers l'autre sur l'écran et se croisent. Au moment où ils se croisent, il est possible qu'ils échangent leurs chapeaux. Il s'agit d'être le premier à dire si oui ou non il y a eu échange des chapeaux lors du croisement.

LE PROGRAMME :
Les lignes 1Ø à 4Ø déterminent les positions des personnages et leurs chapeaux au départ.
La ligne 14Ø teste leur rencontre et la ligne 16Ø saisit "au vol" la réponse d'un des deux joueurs.
La ligne 17Ø teste s'il y a eu réponse.
Les lignes 19Ø à 29Ø éditent le résultat en fonction de la touche pressée.
Les lignes 31Ø et 36Ø constituent le sous-programme d'échange des chapeaux.
Les lignes 37Ø à 44Ø dessinent le bonhomme.

POUR JOUER :
Lancer le programme par RUN.
Les personnages commencent immédiatement à se déplacer l'un vers l'autre.
Le joueur de gauche dispose des touches 1 et 2 et celui de droite, des touches 9 et Ø.
S'il y a eu changement de chapeau, les touches à appuyer seront 1 (à gauche) et Ø (à droite).
S'il n'y a pas eu de changement, il faudra, pour gagner, être le premier à appuyer sur 2 (à gauche) et sur 9 (à droite).
L'affichage indiquera alors BRAVO GAUCHE ou BRAVO DROITE, ou, encore PERDU GAUCHE ou PERDU DROITE.

EXTENSIONS POSSIBLES :
Sur un appareil comme le YENO, il sera possible de prévoir un affichage couleur, avec une animation des personnages.
On pourra aussi compliquer le jeu à plaisir avec plusieurs objets échangés de façon aléatoire (il faudrait alors en trouver la liste complète).

```
10 X1=5
20 C1=162
30 X2=29
40 C2=163
50 CLS
60 X=X1
70 C=C1
80 GOSUB 370
90 X=X2
100 C=C2
110 GOSUB 370
120 X1=X1+1
130 X2=X2-1
140 IF X1<>X2 THEN 160
150 GOSUB 310
160 K$=INKEY$
170 IF K$="" THEN 50
180 GOSUB 310
190 IF ((I<>1)OR(K$<>"1")) AND ((I<>0)
 OR (K$<>"2")) THEN 220
200 PRINT "BRAVO GAUCHE"
210 END
220 IF((I<>1) OR (K$<>"0")) AND ((I<>0
) OR (K$<>"2")) THEN 250
230 PRINT "BRAVO DROITE"
240 END
250 IF((I<>1) OR (K$<>"2")) AND ((I<>0
) OR(K$<>"1")) THEN 280
260 PRINT "PERDU GAUCHE"
270 END
280 PRINT ,"PERDU DROITE"
290 END
300 IF K$="" THEN 50
310 IF RND(1)>.5 THEN 360
320 C=C1
330 C1=C2
340 C2=C
350 I=1
360 RETURN
370 CURSOR X,16:PRINT CHR$(C)
380 CURSOR X,17:PRINT "I"
390 CURSOR X-1,18:PRINT CHR$(144);CHR$
(144);CHR$(144)
400 CURSOR X-1,19:PRINT CHR$(144);CHR$
(144);CHR$(144)
410 CURSOR X-1,20:PRINT CHR$(229);CHR$
(229);CHR$(229)
420 CURSOR X-1,21:PRINT CHR$(229)
430 CURSOR X+1,21:PRINT CHR$(229)
440 RETURN
```

Combat aérien

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexe (1 joueur).
Difficulté du jeu : difficile.

LE JEU :
Au centre de l'écran, se trouve le viseur de votre appareil, et l'avion ennemi se déplace devant vous, d'une façon qui pourra vous sembler au début incohérente.
Vous pouvez vous déplacer dans toutes les directions, grâce aux flèches de mouvement, et vous allez essayer d'amener l'appareil ennemi au centre de votre viseur.
Vous tirerez sur lui en frappant la barre d'espace, et vous voyez vos projectiles qui convergent vers lui. Mais, il continue à se déplacer, même pendant votre tir, et il n'est pas du tout évident que vous réussirez à l'avoir.
La partie continuera à se dérouler jusqu'à ce que vous l'ayez touché (affichage de GAGNE) ou qu'il soit sorti de votre champ de vision.

LE PROGRAMME :
Les lignes 6Ø à 8Ø dessinent l'appareil ennemi dont le déplacement aléatoire sera réalisé par les lignes 11Ø et 12Ø.
Les lignes 14Ø et 17Ø vous permettent le pilotage de votre appareil.
Les lignes 18Ø à 24Ø dessinent le viseur de votre appareil.
Le sous-programme, gérant le tir, va de la ligne 28Ø à la ligne 37Ø et le test de fin de partie est effectué en ligne 35Ø.
La ligne 25Ø regarde si vous avez tiré ou non.

POUR JOUER :
Dès que le programme a été lancé par l'ordre RUN, vous devez manœuvrer pour essayer de faire en sorte que l'appareil ennemi soit cadré dans votre viseur.
Pour cela, vous disposez des flèches de déplacement et vous pouvez tirer à tout moment en pressant la touche d'espace.
L'appareil ennemi peut continuer à se déplacer de manière aléatoire, même lorsque vous avez tiré.
La partie se terminera avec l'affichage GAGNE, si vous touchez l'avion ennemi ou, exceptionnellement, par le fait que l'appareil ennemi sort des limites de l'écran.
On peut aussi arrêter à tout moment en frappant BREAK.

EXTENSIONS POSSIBLES :
Il sera assez facile de faire un affichage au graphisme plus soigné utilisant la couleur. On pourrait aussi envisager de stabiliser la position de l'avion ennemi pendant le tir (il suffit de modifier la ligne 160).
On peut aussi introduire plusieurs avions, compter les points ou le temps, tenir compte de sa propre réserve de carburant, etc.

COMBAT AERIEN

```
10 CLS
20 A=8
30 B=8
40 S=2
50 N=2
60 CURSOR B,A:PRINT CHR$(229)
70 CURSOR B+2,A:PRINT CHR$(229)
80 CURSOR B+1,A:PRINT "-"
90 U=A
100 V=B
110 A=A+S*INT(S*RND(1))-1
120 B=B+S*INT(RND(1)*S)-1
130 K$=INKEY$
140 A=ABS(A+2*(K$=CHR$(31))-(K$=CHR$(3
0)))
150 IF A>22 THEN A=22
160 B=ABS(B+2*(K$=CHR$(28))-(K$=CHR$(2
9)))
170 IF B>33 THEN B=33
180 CURSOR V,U:PRINT "   "
190 CURSOR 16,11:PRINT "¦"
200 CURSOR 18,11:PRINT "/"
210 CURSOR 17,12:PRINT "X"
220 CURSOR 18,12:PRINT " "
230 CURSOR 16,13:PRINT "/"
240 CURSOR 18,13:PRINT "¦"
250 IF K$<>" " THEN 270
260 GOSUB 280
270 GOTO 60
280 CURSOR 5+2*N,24-2*N:PRINT "*"
290 CURSOR 30-2*N,24-2*N:PRINT "*"
300 CURSOR 3+2*N,26-2*N:PRINT " "
310 CURSOR 32-2*N,26-2*N:PRINT " "
320 N=N+1
330 IF N>=7 THEN 350
340 RETURN
350 IF ((A=11) OR (A=12) OR (A=13)) AN
D ((B=16) OR (B=17) OR (B=18)) THEN 37
0
360 GOTO 50
370 CURSOR 0,0:PRINT "GAGNE"
```

Dessin de dés

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : facile.

LE JEU :
Le programme proposé est surtout intéressant par son côté graphique puisqu'il permet de représenter, de façon très convenable, de 1 à 4 dés sur l'écran, en utilisant une technique de dessin un peu particulière.
Chaque réponse à la question NBRE DE DES provoque immédiatement un nouveau lancer du nombre de dés indiqué.

LE PROGRAMME :
Les lignes 4Ø à 12Ø définissent les caractères qui seront utilisés pour les dessins.
La ligne 15Ø permet de contrôler s'il y a eu ou non action sur une touche de 1 à 4.
Les lignes 2ØØ et 21Ø initialisent les deux variables de dessin des dés.
La boucle I (22Ø-41Ø) dessine le cadre du dé.
La ligne 29Ø réalise le tirage, et la boucle J (3ØØ-4ØØ) dessine les points en relation avec le tirage réalisé (en appelant les variables A$ et B$).

POUR JOUER :
On lance le programme par l'ordre RUN, et, à la question NBRE DE DES, on répond en frappant une touche de 1 à 4. Les dés demandés sont alors représentés, et la question NBRE DE DES est de nouveau posée. Il suffira de taper BREAK pour sortir du programme.

EXTENSIONS POSSIBLES :
On pourra conserver la partie dessin utilisée dans ce programme pour réaliser toutes sortes de jeux de dés, avec une représentation graphique de qualité convenable et facile à modifier si besoin était.

DESSIN DE DES

```
10 CLS
20 FF$="FFFFFFFF"
30 FO$="F0F0F0F"
40 PATTERN C# 128,FF$+FF$
50 PATTERN C# 129,"0"+FO$+"0"+FO$
60 PATTERN C# 130,FO$+"0"+FO$+"0"
70 PATTERN C# 131,FF$+FO$+"0"
80 PATTERN C# 132,FO$+"0"+FO$+"0"
90 PATTERN C# 133,FF$+"0"+FO$
100 PATTERN C# 134,"0"+FO$+FF$
110 PATTERN C# 135,FF$+"00000000"
120 PATTERN C# 136,"00000000"+FF$
130 CURSOR 0,0:PRINT "NOMBRE DE DES ?
"
140 K$=INKEY$
150 IF K$="" THEN 140
160 IF ((ASC(K$)<49) OR (ASC(K$)>52))
THEN 140
170 CLS
180 U=8
190 N=ASC(K$)-48
200 A$="      "+CHR$(249)+"  "+CHR$(24
9)+"        "+CHR$(249)+CHR$(249)+"
"+CHR$(249)
210 B$="121314324515525555"
220 FOR I=1 TO N
230 CURSOR U*I-U+2,3:PRINT CHR$(131)
240 CURSOR U*I-U+3,3:FOR J=1 TO 5:PRIN
T CHR$(135);:NEXT J
250 CURSOR U*I-U+8,3:PRINT CHR$(133)
260 CURSOR U*I-U+2,9:PRINT CHR$(132)
270 CURSOR U*I-U+3,9:FOR J=1 TO 5:PRIN
T CHR$(136);:NEXT J
280 CURSOR U*I-U+8,9:PRINT CHR$(134)
290 X=INT(6*RND(1)+1)
300 FOR J=1 TO 5
310 IF J>3 THEN 380
320 S=5*VAL(MID$(B$,3*(X-1)+J,1))-5
330 CURSOR U*I-5,2*J+2:PRINT MID$(A$,S
,1)
340 CURSOR U*I-4,2*J+2:PRINT MID$(A$,S
+1,1)
350 CURSOR U*I-3,2*J+2:PRINT MID$(A$,S
+2,1)
360 CURSOR U*I-2,2*J+2:PRINT MID$(A$,S
+3,1)
370 CURSOR U*I-1,2*J+2:PRINT MID$(A$,S
+4,1)
380 CURSOR U*I-U+2,J+3:PRINT CHR$(130)

390 CURSOR U*I,J+3:PRINT CHR$(129)
400 NEXT J
410 NEXT I
420 GOTO 130
```

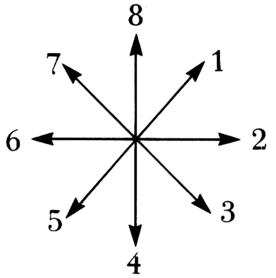
Le jeu du missile

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : contre le YENO.
Difficulté du jeu : moyenne.

LE JEU :
Le YENO est en alerte : vous êtes aux commandes d'un avion chargé d'une mission d'espionnage sur son territoire ; votre but est de survoler une base ennemie (représenté par un H), et de revenir sain et sauf sur votre base ; mais, dès que le YENO vous aura repéré, il va envoyer à votre poursuite, un missile très perfectionné qui n'aura de cesse de vous détruire.
Votre avion peut se déplacer dans toutes les directions, grâce aux touches de 1 à 8 ; réussirez-vous à remplir votre mission ?

LE PROGRAMME :
La ligne 4Ø dessine la base ennemie à survoler (sous forme d'un H).
La ligne 5Ø dessine votre base de départ.
Les lignes 8Ø et 1ØØ choisissent l'instant de départ du missile.
La ligne 12Ø dessine votre avion.
Les lignes 15Ø et 16Ø définissent son mouvement suivant 8 directions, en utilisant les sous-programmes en lignes 28Ø, 29Ø, 31Ø et 32Ø.
La ligne 17Ø assure le départ du missile qui est dessiné par la ligne 18Ø et dont le déplacement est géré par les lignes 19Ø et 2ØØ.
Les lignes 21Ø et 24Ø contrôlent la fin de partie.

POUR JOUER :
Lancer le programme par RUN.
Le terrain de jeu s'affiche et vous pouvez déplacer votre appareil en utilisant les touches de 1 à 8, avec la convention suivante :



Vous devez lui faire contourner le territoire ennemi (si vous, ou le missile, touchez ce territoire, une deuxième partie commence).
Si le missile vous détruit, vous obtenez un message TOUCHE, et la partie s'arrête.
Taper BREAK pour arrêter la partie.

EXTENSIONS POSSIBLES :
Il sera sans doute nécessaire de rajouter divers contrôles dans le programme (position du curseur, survol de la base, retour à sa propre base ...).
Il sera aussi utile de modifier les touches de déplacement en choisissant un groupe de touches pratiques (il faudra alors normaliser leur code avant la ligne 18Ø).
On peut aussi prévoir divers niveaux de jeu en faisant, par exemple, varier la vitesse du missile, prévoir l'utilisation de leurres par l'avion, l'explosion du missile au bout d'un certain temps, etc.

```
    LE JEU DU MISSILE

10 CLS
20 A=2
30 B=5
40 CURSOR 20,B:PRINT "H"
50 CURSOR 3,15:PRINT CHR$(229) ;CHR$(2
29);CHR$(229)
60 U=15
70 X=14
80 S=INT(U*RND(1))+1
90 Y=1
100 S=S-1
110 V=29
120 CURSOR Y+3,X:PRINT "+"
130 K$=INKEY$
140 IF K$="" THEN 130
150 F=-A*(ASC(K$)-48<B)-1
160 ON ASC(K$)-48+A*(F-1) GOSUB 280,29
0,310,320
170 IF S<>0 THEN 100
180 CURSOR V+3,U:PRINT "<"
190 U=U-SGN(U-X)
200 V=V-SGN(V-Y)
210 IF (U-X<>0) OR (Y-17<>0) THEN 240
220 CURSOR 0,0:PRINT ,"TOUCHE....."
230 END
240 IF (U=B) AND (V=17) THEN 10
250 IF (X-B<>0) OR (Y-17<>0) THEN 120
260 CURSOR 0,0:PRINT "MISSION ACCOMPLI
E..."
270 END
280 X=X-F
290 Y=Y+F
300 RETURN
310 GOSUB 290
320 X=X+F
330 RETURN
```

Le singe

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : deux joueurs.
Difficulté du jeu : difficile.

LE JEU :
Il s'inspire de l'idée déjà ancienne du singe qui, frappant au hasard sur une machine à écrire, a des chances non nulles de finir par écrire un texte signifiant.
Le rôle du singe est, ici, tenu par le YENO qui va afficher des groupes de 3 à 6 lettres pendant un temps relativement court.
Si l'un des joueurs estime qu'il peut réussir à faire un mot avec les lettres proposées, il doit taper rapidement sur sa touche de jeu (1 ou Ø).
Le mot devra alors être accepté ou refusé par l'adversaire qui répondra JUSTE ou FAUX.
Chaque mot juste rapporte 1 point, chaque mot faux enlève 1 point.

LE PROGRAMME :
La ligne 4Ø contient une liste V$ des voyelles courantes, et la ligne 5Ø, une liste C$ des consonnes courantes.
Il sera possible de modifier ces deux listes pour augmenter, par exemple, la fréquence d'apparition de certaines lettres (on pourra ainsi mettre deux E dans V$ ou deux R dans C$, etc.).
La ligne 6Ø définit la longueur du mot (en principe, de 3 à 6 lettres, mais on peut également modifier cette longueur).
La boucle I (8Ø-15Ø) permet la confection d'un “mot” en prenant une lettre, soit dans l'alphabet complet, soit dans la liste C$ (ligne 11Ø), soit dans la liste V$ (ligne 13Ø).
La ligne 17Ø temporise l'apparition du résultat.
La ligne 2ØØ gère la fin de partie, avec affichage du score.
La ligne 23Ø regarde si l'un des joueurs a réagi.
Enfin, les lignes 25Ø et 26Ø comptabilisent les points de chaque joueur.

POUR JOUER :
On lancera le programme en tapant RUN.
La première série de lettres est alors affichée sur l'écran. Le joueur de gauche devra taper 1 s'il voit qu'il peut réaliser un mot, celui de droite devra taper Ø.
Si rien n'a été frappé, une autre série de lettres sera affichée après très peu de temps.
Si l'un des joueurs a frappé Ø ou 1, la question (J) ou (F) est alors affichée.
On répond J si le mot est correct, F si le mot est incorrect, suivi de CR.
Les points sont alors calculés par YENO, et la série de lettres suivante est alors affichée.
Pour sortir du programme, il faudra taper P au lieu de Ø ou 1 ; le score final est alors affiché.

EXTENSIONS POSSIBLES :

On pourra modifier la longueur des mots, la fréquence d'apparition des lettres, le décompte des points, la règle du jeu, etc.

LE SINGE

```
10 CLS
20 A=0
30 B=A
40 V$="AEIOU"
50 C$="BCDGLMNPRSTV"
60 X=INT(4*RND(1)+3)
70 M$=""
80 FOR I= 1 TO X
90 E$=CHR$(INT(26*RND(1)+65))
100 IF RND(1)>.4 THEN 120
110 E$=MID$(C$,INT(LEN(C$)*RND(1))+1,1
)
120 IF RND(1)>.5 THEN 140
130 E$=MID$(V$,INT(LEN(V$)*RND(1))+1,1
)
140 M$=M$+E$
150 NEXT I
160 PRINT M$
170 FOR P=1 TO 500
180 NEXT P
190 K$=INKEY$
200 IF K$<>"P" THEN 230
210 PRINT :PRINT "SCORE A ";A,"SCORE B
 ";B
220 END
230 IF (K$<>"0") AND (K$<>"1") THEN 60

240 INPUT "(J)USTE OU (F)AUX ?";R$
250 A=A+(R$="J")*(K$="1")-(K$="1")*(R$
="F")
260 B=B+(R$="J")*(K$="0")-(K$="0")*(R$
="F")
270 GOTO 60
```

Le jeu du 15

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : moyenne.

LE JEU :
C'est un jeu pour deux joueurs, assez ancien, qui était autrefois pratiqué dans les foires. On dispose d'un tapis de jeu de 9 cases, et chacun mise à son tour un jeton de sa couleur, sur une case disponible.
Le premier des deux joueurs qui, avec exactement 3 jetons, arrive à faire un total de 15, a gagné.

LE PROGRAMME :
La boucle I (11Ø-16Ø) dessine le terrain de jeu, et la ligne 19Ø s'occupe de la gestion du tour des joueurs (numéro 1 ou 2 pour chaque joueur, et jeton O ou X correspondant).
La ligne 23Ø attend le coup joué.
La boucle I (25Ø-28Ø) vérifie la validité du coup.
La ligne 29Ø teste la fin de partie, en cas de match nul, et la ligne 32Ø refuse les coups non conformes (avec retour à la situation initiale, ligne 18Ø).
Les lignes 34Ø et 38Ø enregistrent les coups respectifs de chaque joueur dans des variables alphanumériques O$ et X$.
Les lignes 35Ø et 37Ø s'occupent de l'affichage du coup, et la ligne 39Ø l'enregistre dans une variable R$. La partie la plus intéressante du programme, qui pourrait servir à définir une stratégie machine, va de la ligne 4ØØ à la ligne 53Ø : on utilise 3 boucles emboîtées pour vérifier si l'un des joueurs a obtenu, avec 3 pions, un total de 15. Le changement de joueur s'effectue en ligne 54Ø.

POUR JOUER :
Lancer le jeu en tapant RUN.
Le terrain de jeu s'affiche, puis la question JOUEUR 1 : X, et le joueur, qui a choisi les X, devra donner le numéro de la case où il désire jouer ; si elle est libre, son coup y sera enregistré.
On obtient, ensuite, l'affichage JOUEUR 2 : O et le deuxième joueur procède de la même façon. Si l'un des joueurs gagne (total de 15 sur 3 coups), il obtient l'affichage BRAVO et la partie s'arrête.
En cas de match nul, la machine affiche FINI.
On fera une autre partie en tapant RUN.

EXTENSIONS POSSIBLES :

Le plus intéressant sera sans doute d'essayer de définir une stratégie machine (en essayant d'éviter que la machine ne joue toujours le 5 lorsqu'elle doit commencer).

On peut aussi, bien sûr, améliorer le graphisme et introduire des effets sonores pour les différents messages.

```
LE JEU DU 15

10 CLS
20 F1$="FFFFFFFF"
30 F2$="F0F0F0F"
40 PATTERN C# 128,F2$+"0"+F1$
50 PATTERN C# 129,"00000000"+F1$
60 PATTERN C# 130,"0"+F2$+F1$
70 F=0
80 R$="123456789"
90 O$=""
100 X$=""
110 FOR I=1 TO 24 STEP 3
120 CURSOR I+3,15:PRINT CHR$(128)
130 CURSOR I+4,15:PRINT CHR$(129)
140 CURSOR I+5,15:PRINT CHR$(130)
150 CURSOR I+4,16:PRINT CHR$(49+I/3)
160 NEXT I
170 PRINT :PRINT TAB(8);"JOUEUR :":PRI
NT:PRINT:PRINT :PRINT
180 CURSOR 17,18 :PRINT CHR$(49+F)
190 IF F<>0 THEN 220
200 CURSOR 20,18:PRINT "X"
210 GOTO 230
220 CURSOR 20,18:PRINT "0"
230 K$=INKEY$: IF K$="" THEN 230
240 IF (ASC(K$)<49) OR (ASC(K$)>57) TH
EN 230
250 FOR I=1 TO LEN(R$)
260 IF K$<>MID$(R$,I,1) THEN 280
270 GOSUB 330
280 NEXT I
290 IF VAL(R$)<>0 THEN 320
300 CURSOR 5,0 :PRINT "FINI"
310 GOTO 560
320 GOTO 180
330 IF F<>0  THEN 370
340 X$=X$+K$
350 CURSOR 3*I+2,14:PRINT "X"
360 GOTO 390
370 CURSOR 3*I+2,14:PRINT "0"
380 O$=O$+K$
390 R$=LEFT$(R$,I-1)+"0"+RIGHT$(R$,LEN
(R$)-I)
400 IF F<>0  THEN 430
410 T$=X$
420 GOTO 440
430 T$=O$
440 IF LEN(T$)<3  THEN 540
450 FOR II=1 TO 3
460 FOR J=II+1 TO LEN(T$)
470 FOR K=J+1 TO LEN(T$)
480 IF VAL(MID$(T$,II,1))+VAL(MID$(T$,
J,1))+VAL(MID$(T$,K,1))<>15  THEN 510
490 CURSOR 5,0:PRINT "BRAVO !!!"
500 GOTO 560
510 NEXT K
520 NEXT J
530 NEXT II
540 F=ABS(F-1)
550 RETURN
560 END
```

Adresse

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexe (pour 1 joueur).
Difficulté du jeu : difficile.

LE JEU :
Des chiffres, entre Ø et 9, vont apparaître au hasard, en divers emplacements de l'écran. Ils restent affichés pendant un temps très court. Le joueur doit réussir à appuyer sur la touche numérique correspondante, au moment où le chiffre est affiché.
Le chiffre tapé par le joueur sera alors affiché en grand format sur la gauche de l'écran ; et chaque réponse exacte rapporte 1Ø points. La durée de la partie est aléatoire et le score sera affiché en fin de partie.

LE PROGRAMME :
Les lignes 1ØØ et 11Ø contiennent, dans deux variables alphanumériques C$ et B$, les références qui seront utilisées pour dessiner les chiffres en grand format.
L'emplacement du chiffre tiré par le YENO est choisi aux lignes 12Ø et 13Ø, et le chiffre est tiré en ligne 14Ø.
La ligne 15Ø réalise l'impression, et la ligne 16Ø détermine la durée d'affichage. La ligne 2ØØ définit la durée du jeu, et la fin de partie est traitée par la ligne 22Ø.
La boucle I (29Ø-33Ø) réalise le dessin grand format du chiffre frappé par le joueur.
Le contrôle de l'exactitude du résultat est effectué par la ligne 34Ø.

POUR JOUER :
Le programme sera lancé par RUN, puis un premier chiffre apparaît, après une courte pause, vers le milieu de l'écran. Le joueur devra réagir suffisamment vite, tant qu'il reste à l'affichage, et le retaper. Le chiffre qu'il a tapé se dessinera alors en haut à gauche de l'écran, en grand format.
Si le résultat est exact, le joueur marquera 1Ø points.
Le temps de déroulement de la partie sera fonction des chiffres qui apparaîtront à l'affichage.
En fin de partie, le score sera affiché. Il sera possible de s'arrêter à tout moment en tapant BREAK.

EXTENSIONS POSSIBLES :
On peut envisager de modifier le jeu pour faire participer plusieurs joueurs, en modifiant éventuellement l'affichage et en codant les touches du clavier mises à la disposition de chacun des joueurs.
On pourra aussi enlever des points à chaque résultat faux, modifier les règles de calcul du score et la durée du jeu, etc.

```
10 M=10
20 T=10
30 CLS
40 FOR I=1 TO 200
50 NEXT I
60 S=0
70 PATTERN C# 128, "FFFFFFFFFFFFFFFF"
80 PATTERN C# 129, "F0F0F0F0F0F0F0F0"
90 PATTERN C# 130, "0F0F0F0F0F0F0F0F"
100 C$=CHR$(128)+CHR$(128)+CHR$(129)+"
  "+CHR$(130)+CHR$(129)+CHR$(130)
110 B$="144413333313121131314413312131
12141133331414114131"
120 X=INT(35*RND(1))+3
130 Y=INT(19*RND(1))+4
140 N=INT(M*RND(1))
150 CURSOR X,Y:PRINT CHR$(48+N)
160 FOR P=1 TO 200+200*RND(1)
170 NEXT P
180 K$=INKEY$
190 IF K$<>"" THEN 270
200 T=T-1
210 CURSOR X,Y:PRINT " "
220 IF T>0 THEN 120
230 CLS
240 CURSOR 0,20:PRINT "SCORE ";S;"/100
"
250 K$=INKEY$:IF K$="" THEN 250
260 GOTO 10
270 U=ASC(K$)-48
280 BEEP
290 FOR I=1 TO 5
300 A=2*VAL(MID$(B$,5*U+I,1))-1
310 CURSOR 3,I:PRINT MID$(C$,A,1)
320 CURSOR 4,I:PRINT MID$(C$,A+1,1)
330 NEXT I
340 IF U-N<>0 THEN 360
350 S=S+M
360 GOTO 200
```

Sous-marin

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexes (1 seul joueur).
Difficulté du jeu : facile.

LE JEU :
Vous manœuvrez en surface un chasseur de sous-marins, et des sous-marins passent très régulièrement en dessous de votre unité, à différentes profondeurs.
Vous pouvez, à tout moment, larguer des grenades sous-marines et vous pouvez même envisager de toucher plusieurs fois le même sous-marin.
Votre score sera affiché en permanence, sur la gauche de l'écran (rapport nombre de coups au but sur nombre de sous-marins apparus).

LE PROGRAMME :
La ligne 11Ø permet au YENO de choisir la profondeur du sous-marin.
Les lignes 15Ø à 25Ø dessinent votre bâtiment.
La ligne 12Ø affiche le score et la ligne 21Ø dessine la surface de l'eau.
Les lignes 22Ø à 25Ø dessinent le sous-marin.
Le tir est effectué par un programme spécifique placé aux lignes 39Ø à 43Ø.
Le déplacement (très rapide) de votre navire est réalisé par la ligne 37Ø.
Le test, permettant de savoir si l'on a touché ou non, est effectué en ligne 28Ø.

POUR JOUER :
Taper RUN pour commencer la partie.
Le bâtiment sera déplacé grâce aux flèches droite et gauche.
La touche d'espace permet le tir.
Le navire, se déplaçant trois fois plus vite que les sous-marins, il sera possible de toucher plusieurs fois le même bâtiment.
Pour terminer la partie, il conviendra de faire BREAK.

EXTENSIONS POSSIBLES :
Il sera assez élémentaire d'améliorer la qualité graphique du jeu, d'utiliser la couleur et les sons, de faire apparaître plusieurs sous-marins se déplaçant à des vitesses différentes, éventuellement même, de permettre aux sous-marins de tirer...

```
10 F$="FFFFFFFF"
20 0$="00000000"
30 PATTERN C# 128,F$+F$
40 PATTERN C# 129,0$+F$
50 PATTERN C# 130,F$+0$
60 C=12
70 S=3
80 P=.01
90 CLS
100 T=34
110 S=INT(6*RND(1)+7)
120 PRINT ,"SCORE";P
130 P=P+.01
140 Y=5
150 CURSOR C+3,4:PRINT "    ":CURSOR C-
3,4:PRINT "    "
160 CURSOR C+6,4:PRINT CHR$(129);CHR$(
129)
170 CURSOR C+8,4:PRINT CHR$(128)
180 CURSOR C+9,4:PRINT CHR$(129)
190 CURSOR C+10,4:PRINT "/"
200 CURSOR C+11,4:PRINT"    "
210 CURSOR 3,5:FOR J=1 TO 35:PRINT CHR
$(130);:NEXT J
220 CURSOR T,S:PRINT CHR$(129)
230 CURSOR T+1,S:PRINT CHR$(128)
240 CURSOR T+2,S:PRINT CHR$(129)
250 CURSOR T+3,S:PRINT " "
255 PRINT C
260 IF Y<>5 THEN 280
270 X=C+Y+Y-2
280 IF (S<>Y) OR (T+2<>X) THEN 310
290 P=P+1
300 CURSOR T,S:PRINT "***"
310 T=T-1
320 IF P<>99 THEN 350
330 PRINT "C'EST FINI.. "
340 END
350 IF T<3 THEN 90
360 K$=INKEY$
370 C=C+3*(K$=CHR$(28))*(C<21)-3*(K$=C
HR$(29))*(C>3)
380 IF (K$<>" ") AND (Y=5) THEN 150
390 CURSOR X,Y:PRINT " "
400 Y=Y+1
410 IF Y>S THEN 140
420 CURSOR X,Y:PRINT "0"
430 GOTO 150
```

Zombies

REFERENCES :
Difficulté de programmation : assez difficile.
Catégorie de jeu : solitaire.
Difficulté du jeu : assez facile.

LE JEU :
Les ZOMBIES sont des êtres malfaisants qui vont essayer de vous détruire, mais vous allez, vous aussi, essayer de les utiliser à votre profit, pour débarrasser votre espace vital des trous noirs qui l'obscurcissent. Sur l'écran, est tracé un cadre qui définit votre espace vital. Dans ce cadre, vont apparaître de façon aléatoire, les ZOMBIES (ce sont des Z), les trous noirs (des carrés noirs) et le joueur (un O). Les ZOMBIES convergent vers le joueur pour essayer de le capturer, mais celui-ci se déplace plus rapidement qu'eux, et il va essayer de leur faire avaler tous les trous noirs, sans se faire prendre.
La partie se termine, soit avec la disparition du joueur, soit par l'élimination de tous les trous noirs.

LE PROGRAMME :
La position des ZOMBIES est définie par un tableau Z.
Les lignes 3Ø à 12Ø (boucle I) dessinent le cadre, les ZOMBIES et les trous noirs.
La ligne 21Ø définit la position initiale du joueur.
Une deuxième boucle I (15Ø-2ØØ) réalise le déplacement des Z.
Les lignes 26Ø et 27Ø contrôlent le déplacement du joueur.

POUR JOUER :
Après avoir lancé l'exécution par RUN, le joueur devra se déplacer en utilisant les 4 flèches et en veillant à ne pas sortir des limites du terrain.
Il devra manœuvrer de façon à faire venir les ZOMBIES sur les trous noirs, sans se faire lui-même coincer.
Pour arrêter la partie, il suffira de taper BREAK.

EXTENSIONS POSSIBLES :
Il serait intéressant d'agrandir le terrain de jeu, d'augmenter le nombre de ZOMBIES ou de trous noirs, d'améliorer les contrôles (pratiquement inexistants), de placer le joueur de façon plus aléatoire, éventuellement de modifier la règle (par exemple faire disparaître les ZOMBIES dans les trous noirs).
On pourra aussi, bien sûr, améliorer la qualité du dessin et introduire la couleur.

ZOMBIES

```
10 CLS
20 L=3
30 FOR I=L TO 21
40 CURSOR I+1,L:PRINT CHR$(229)
50 CURSOR L,I:PRINT CHR$(229)
60 CURSOR 23,I:PRINT CHR$(229)
70 IF RND(1)>.5 THEN 90
80 CURSOR I,3+INT(19*RND(1)):PRINT CHR
$(229)
90 S=INT(14*RND(1))+4
100 Z((I-1)/2)=S
110 CURSOR I+1,21:PRINT CHR$(229)
120 NEXT I
130 K=INT(18*RND(1))+4
140 L=2
150 FOR I=1 TO 4
160 CURSOR Z(I+4),3+Z(I):PRINT " "
170 Z(I)=Z(I)-SGN(Z(I)+3-S)
180 Z(I+4)=Z(I+4)-SGN(Z(I+4)-K)
190 CURSOR Z(I+4),3+Z(I):PRINT "Z"
200 NEXT I
210 CURSOR K,S:PRINT "0"
220 FOR P=1 TO 250
230 NEXT P
240 CURSOR K,S:PRINT " "
250 K$=INKEY$
260 K=K+L*(K$=CHR$(29))-L*(K$=CHR$(28)
)
270 S=S+L*(K$=CHR$(30))-L*(K$=CHR$(31)
)
280 GOTO 150
```

Diago

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : réflexes (1 seul joueur).
Difficulté du jeu : difficile.

LE JEU :
On lance le programme par RUN et on obtient un premier affichage qui vous renseigne sur la performance à réaliser (par exemple, 5 points noirs ou 17 points noirs...), puis le temps est affiché et décompté, à partir de 1Ø secondes. Le jeu commence lorsque l'on arrive à Ø seconde. Des points noirs, apparaissent alors en bas et à gauche de l'écran, formant une ligne diagonale, et il vous faudra appuyer sur une touche quelconque pour arrêter l'affichage des points dès que vous estimez que le nombre de points demandés au début est apparu à l'écran.
On obtient alors un affichage BRAVO ou PERDU selon le résultat obtenu.

LE PROGRAMME :
Le nombre de points noirs sera choisi au hasard, en ligne 2Ø.
Le temps est décompté par les lignes 8Ø à 12Ø.
Les points sont imprimés par la ligne 16Ø.
La pression d'une touche est détectée en ligne 2ØØ, et la ligne 21Ø contrôle la fin de partie (aucun réflexe).
Les affichages de fin sont exécutés par les lignes 24Ø à 29Ø.

POUR JOUER :
On lancera l'exécution par RUN.
On obtient alors l'affichage de la phrase LE BON REFLEXE EST A... qui indique le nombre de points gagnants.
Le temps sera ensuite décompté sous la forme ATTENTION ... SECONDES qui indique le temps restant avant l'apparition de la diagonale.
A l'affichage de Ø, la diagonale apparaît et commence à se déplacer sur l'écran.
Si l'on estime que le nombre de points réclamés a été atteint, on devra presser la touche Ø.
L'affichage indique alors BRAVO si l'on a gagné, ou PERDU dans le cas contraire ; puis la phrase AUTRE ESSAI (O ou N) ? est affichée, répondre O pour faire une autre partie.

EXTENSIONS POSSIBLES :
On pourra envisager une adaptation du jeu à deux joueurs ou plus, en laissant la diagonale se former complètement et en testant la coïncidence lors de la pression de la touche affectée au joueur.
On peut aussi réaliser un graphisme plus soigné et modifier le principe du déplacement des objets sur l'écran (déplacement horizontal, vertical ou apparition au hasard...).

DIAGO

```
10 CLS
20 X=INT(3+18*RND(1))
30 PRINT "LE BON REFLEXE EST A";X:PRIN
T :PRINT :PRINT :PRINT
40 FOR P=1 TO 300
50 NEXT P
60 S=10
70 PRINT "ATTENTION :   SECONDES..":PR
INT :PRINT
80 S=S-1
90 CURSOR 12,5:PRINT CHR$(48+S)
100 FOR P=1 TO 100
110 NEXT P
120 IF S<>0 THEN 80
130 CLS
140 U=22
150 V=36
160 CURSOR V,U:PRINT "*":FOR Y=1 TO 25
:NEXT Y
170 U=U-1
180 V=V-1
190 K$=INKEY$
200 IF K$<>"" THEN 230
210 IF U+X<>22 THEN 160
220 GOTO 310
230 IF U+X<>22 THEN 310
240 PRINT :PRINT :PRINT ,"BRAVO"
250 FOR T=1 TO 1000
260 NEXT T
270 CLS
280 INPUT "AUTRE ESSAI (O/N) ? ";R$
290 IF R$="O" THEN 10
300 END
310 CURSOR 0,20:PRINT "PERDU"
320 GOTO 250
```

Topologie

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : graphisme.
Difficulté du jeu : moyenne.

LE JEU :
Un dessin, très élémentaire, à base de traits apparaît sur l'écran (il peut aussi être entré par un partenaire).
Le joueur dispose d'un plot mobile symbolisant une gomme et devra essayer d'effacer le dessin précédent en repassant, une fois et une seule, sur chacun des traits.

LE PROGRAMME :
La ligne 2Ø contient un dessin préprogrammé (variable B$). On pourra, bien entendu, soit définir un autre dessin, soit entrer le dessin en cours de jeu en tapant 2Ø INPUT B$.
Le codage du dessin est réalisé de la façon suivante :

```
Début                                    ┌─Longueur du déplacement
          → x x | | x x | | | x x ¦ x x | | x x | x x | | ... etc.
                              │ └──Déplacement en y (Ø, 1 ou 2)
            x     y           │
                              └────Déplacement en x (Ø, 1 ou 2)

      Position de
        départ

     Ø    ↑              Ø    ←
 x : 1  immobile     y : 1  immobile
     2    ↓              2    →
```

Les lignes 3Ø et 4Ø calculent la position de départ.
La boucle I (5Ø-9Ø) assure le décodage, l'impression étant assurée par le sous-programme 19Ø-24Ø. La partie effacement commence en ligne 1ØØ.
Le décodage du déplacement du joueur est assuré par les lignes 16Ø et 17Ø (8 directions).

```
        E               W   R
        ↑                ↖ ↗
   S←       →D
        ↓                ↙ ↘
        X               Z   C
```

```
10 CLS
20 B$="0204221120111218001102111018"
30 S=VAL(MID$(B$,1,2))
40 T=VAL(MID$(B$,3,2))
50 FOR I=5 TO LEN(B$) STEP 4
60 U=VAL(MID$(B$,I,1))-1
70 V=VAL(MID$(B$,I+1,1))-1
80 GOSUB 190
90 NEXT I
100 X=1
110 Y=3
120 CURSOR Y,X:PRINT CHR$(229)
130 CURSOR Y,X:PRINT " "
140 K$=INKEY$
150 IF K$="" THEN 120
160 Y=Y+(K$="W")+(K$="Z")+(K$="S")-(K$
="D")-(K$="C")-(K$="R")
170 X=X+(K$="E")+(K$="R")+(K$="W")-(K$
="C")-(K$="X")-(K$="Z")
180 GOTO 120
190 FOR K=1 TO VAL(MID$(B$,I+2,2))
200 CURSOR S,T:PRINT CHR$(229)
210 S=S+U
220 T=T+V
230 NEXT K
240 RETURN
```

Chasse au sous-marin

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de réflexion.
Difficulté du jeu : moyenne.

LE JEU :
Vous êtes à bord d'un bâtiment de surface, chargé de traquer un sous-marin caché quelque part dans votre secteur ; vous devez vous déplacer jusqu'à venir au-dessus du sous-marin et tirer à une certaine profondeur ; mais, attention : le sous-marin, lui aussi, est mobile ; il ne coupera ses moteurs que lorsque vous serez exactement à sa verticale, pour ne pas être repéré.

LE PROGRAMME :
La boucle I (7Ø-1ØØ) affiche le terrain.
Les lignes 14Ø et 15Ø définissent la position initiale du sous-marin ; les lignes 16Ø et 17Ø permettent son déplacement, contrôlé par la ligne 18Ø (on reste dans les limites du terrain).
Les lignes 27Ø et 28Ø contrôlent votre déplacement.
La distance est calculée en ligne 29Ø, affichée en ligne 3ØØ et testée en ligne 32Ø.
Le test de fin de partie est effectué à la ligne 41Ø.

POUR JOUER :
On lance le programme par RUN et on entre la dimension ; la grille de jeu s'affiche. Vous êtes au départ en position (O, O) et vous vous déplacez avec les touches de déplacement.
Après chaque déplacement, la nouvelle distance du sous-marin s'affiche ; à chaque coup le sous-marin se déplace. Si vous réussissez à vous placer à sa verticale, la distance Ø est affichée et il ne bouge plus.
Vous tirez en donnant une profondeur entière (entre 2 et 4, bornes comprises). Si vous le ratez, il se replace n'importe où, et on recommence ; si vous le coulez, vous obtenez l'affichage BRAVO et la partie s'arrête.
On peut arrêter à tout moment le programme en utilisant la touche BREAK.

EXTENSIONS POSSIBLES :
Il est possible de prévoir une amélioration du graphique, d'améliorer les possibilités de déplacement du chasseur (un peu lent, dans cette version), de prévoir des parcours plus sophistiqués pour le sous-marin, de mettre un compteur de temps...

POUR JOUER :

On commence avec l'instruction RUN et, si le dessin n'a pas été programmé, on le rentre en suivant les instructions données précédemment. On tape ensuite CR.

A l'affichage du dessin, un point lumineux clignotant apparaît en bas à gauche de l'écran. On peut le déplacer en utilisant les 4 touches de déplacement d'une part, et en utilisant les touches pour les déplacements en diagonale d'autre part.

On devra essayer d'effacer le dessin initial, sans lever le crayon et sans repasser deux fois sur le même trait.

EXTENSIONS POSSIBLES :

On pourra introduire la couleur, définir une table de dessins topologiquement intéressants, ajouter des contrôles sur le fait que l'on ne quitte pas le dessin initial, etc.

```
      CHASSE AU SOUS-MARIN


10 CLS
20 A=0
30 INPUT "DIMENSION (max 9) :";H
40 CLS
50 PRINT "DISTANCE"
60 G=2
70 FOR I=0 TO H
80 CURSOR 5,I+9:PRINT CHR$(48+I)
90 CURSOR 8+I*2,7:PRINT CHR$(48+I)
100 NEXT I
110 I=1
120 X=7
130 Y=2
140 V=INT(H*RND(1))+7
150 U=INT(H*RND(1))+2
160 U=U+I
170 V=V+I
180 IF (U>=2) AND (U<=H+2) AND (V>=7)
AND (V<=H+7)  THEN 200
190 I=-I
200 CURSOR 2+G*Y,X+G:PRINT "  "
210 CURSOR 4+G*Y,X+G:PRINT CHR$(229)
220 CURSOR 5+G*Y,X+G:PRINT "  "
230 CURSOR 4+G*Y,X+1:PRINT " "
240 CURSOR 4+G*Y,X+3:PRINT " "
250 K$=INKEY$
260 IF K$="" THEN 250
270 Y=Y+(K$=CHR$(28))*(Y<H+2)-(K$=CHR$
(29))*(Y>2)
280 X=X+(K$=CHR$(31))*(X<H+7)-(K$=CHR$
(30))*(X>7)
290 D=INT(SQR((X-V)*(X-V)+(Y-U)*(Y-U))
)
300 CURSOR 12,0:PRINT D
310 IF D=0 THEN 340
320 IF D<3 THEN 160
330 GOTO 200
340 CURSOR 22,0:PRINT "P"
350 CURSOR 24,0:PRINT "?"
360 BEEP
370 K$=INKEY$
380 IF K$="" THEN 370
390 CURSOR 22,0:PRINT "   "
400 D=VAL(K$)
410 IF (D-INT(3*RND(1))-2)<>0 THEN 150

420 PRINT "BRAVO !!"
430 END
```

Le lézard d'Alice

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : solitaire.
Difficulté du jeu : assez difficile.

LE JEU :
Il s'inspire très vaguement d'une idée de LEWIS CARROLL, dans "Alice au Pays des Merveilles". Dans l'histoire originale, Alice grandissait ou diminuait, alors que dans le jeu, c'est un lézard qui change de taille.
Deux verres apparaissent sur l'écran : l'un contient le liquide qui fait grandir, l'autre le liquide qui fait rapetisser (malheureusement ce n'est pas toujours le même verre). Le joueur devra indiquer la quantité de liquide qu'il désire boire et le lézard changera de taille en fonction de cette quantité. Le but du jeu est de parvenir à lui rendre sa taille initiale.

LE PROGRAMME :
Les lignes 4Ø à 9Ø dessinent les verres.
La boucle I (15Ø-17Ø) les remplit avec la quantité indiquée en ligne 12Ø.
La variation de taille du lézard est définie par la ligne 18Ø, avec contrôle par les lignes 19Ø et 21Ø.
La ligne 23Ø dessine le corps du lézard, les lignes 24Ø à 26Ø dessinent l'œil et les pattes.
Un message gagnant sera affiché par la ligne 3ØØ si l'on a trouvé la bonne taille.

POUR JOUER :
On lance le programme par RUN : les deux verres se dessinent. A la question VERRE ? on répond en tapant G ou D suivi de CR.
A la question COMBIEN ? on entre alors une quantité (de Ø à 9) suivi de CR.
Le verre désigné se remplit sous vos yeux, puis, après quelques instants, le lézard est affiché. On devra alors frapper une touche quelconque (sauf BREAK qui arrêterait la partie) ; si le lézard n'avait pas la bonne taille, aucun message n'est affiché et l'on revient au dessin des verres.

EXTENSIONS POSSIBLES :
Puisqu'il s'agit d'un jeu essentiellement graphique, il sera intéressant d'améliorer les dessins, d'introduire la couleur, de modifier le caractère aléatoire du choix des verres, etc.
Pourquoi ne pas relire Alice au Pays des Merveilles, les idées n'y manquent pas.

LE LEZARD D'ALICE

```
10 PATTERN C# 128,"FFFFFFFF00000000"
20 N=INT(15*RND(1))+1
30 CLS
40 CURSOR 23,10:FOR J=1 TO 9:PRINT CHR
$(229);:NEXT J
50 CURSOR 9,10:FOR J=1 TO 9:PRINT CHR$
(229);:NEXT J
60 CURSOR 18,10:FOR J=1 TO 9:PRINT CHR
$(229);:NEXT J
70 CURSOR 2,10:FOR J=1 TO 9:PRINT CHR$
(229);:NEXT J
80 CURSOR 9,20:FOR J=1 TO 5:PRINT CHR$
(128);:NEXT J
90 CURSOR 19,20:FOR J=1 TO 5:PRINT CHR
$(128);:NEXT J
100 PRINT :PRINT TAB(9);"GAUCHE";TAB(1
9);"DROITE"
110 CURSOR 0,0:INPUT "        VERRE   :
";V$
120 INPUT "        COMBIEN :";Q
130 PRINT :PRINT :PRINT :PRINT
140 C=-10*(V$="G")-20*(V$="D")
150 FOR I=20-Q TO 19
160 CURSOR C,I:PRINT CHR$(144);CHR$(14
4);CHR$(144)
170 NEXT I
180 N=N+Q*2*((RND(1)>.5)+.5)
190 IF N<=30 THEN 210
200 N=30
210 IF N>=1 THEN 230
220 N=1
230 CURSOR 17-N/2,8:FOR J=1 TO N:PRINT
 CHR$(229);:NEXT J
240 CURSOR 17-N/3,9:PRINT "^"
250 CURSOR 17+N/3,9:PRINT "^"
260 CURSOR 16+N/2,7:PRINT "@"
270 K$=INKEY$
280 IF K$="" THEN 270
290 IF N<>10 THEN 30
300 CURSOR 25,4:PRINT "BRAVO"
```

Calculatrice

REFERENCES :
Difficulté de programmation : très difficile.
Catégorie de jeu : simulation d'une calculatrice.
Difficulté du jeu : moyenne.

LE JEU :
Il s'agit de la simulation, en affichage grand format, du fonctionnement d'une petite calculatrice de poche, de type 4 opérations. Les applications pédagogiques de ce programme sont évidentes et son fonctionnement très simple à maîtriser.

LE PROGRAMME :
Les variables B$ et C$, des lignes 2Ø et 3Ø, contiennent les informations codées qui permettront le dessin des chiffres en grand format.
Les lignes 9Ø à 12Ø testent les différentes entrées possibles.
Les lignes 22Ø à 26Ø s'occupent de la représentation, en grand format, des chiffres sur l'écran, au fur et à mesure des entrées.
Le point décimal et le signe – sont gérés par les lignes 17Ø et 3ØØ.
Le nombre total des symboles entrés est contrôlé par la ligne 2ØØ.
Le résultat est inscrit grâce à un programme fonctionnant en automatique qui se trouve aux lignes 4ØØ à 57Ø (l'automatisme est géré par la ligne 1ØØ, dans le programme principal).

POUR JOUER :
On devra lancer le programme par RUN.
Puis, on tape les chiffres du premier nombre à entrer qui vont s'afficher un par un, au fur et à mesure (on ne pourra entrer plus de 7 symboles, y compris le signe – et le point décimal). La frappe d'un signe opératoire laisse l'affichage intact, mais l'introduction du prochain chiffre ou symbole provoque l'effacement immédiat et l'affichage du nouveau symbole.
Il est à remarquer que les chiffres s'introduisent de gauche à droite, sens de l'écriture usuelle.
L'appareil accepte, comme signes opératoires, les symboles +, –, *, / ; le signe = provoquera l'affichage du résultat. Les opérations sont enchaînées automatiquement, le résultat précédent étant pris en compte comme opérateur du calcul suivant.
Pour faire une autre opération, il faudra taper BREAK puis RUN.
Taper BREAK pour sortir du programme.

EXTENSIONS POSSIBLES :
On devra, tout d'abord, améliorer l'affichage du signe – (représenté, ici, par un seul caractère), puis on pourra améliorer le dessin des chiffres ; on peut aussi envisager une extension à 8 chiffres, autoriser des opérations plus complexes (COS, SIN, ...), ajouter des mémoires, etc.

CALCULATRICE

```
10 CLS
20 B$="1444133333131211313144133121311
2141133331414114131"
30 C$=CHR$(229)+CHR$(229)+CHR$(224)+"
 "+CHR$(227)+CHR$(224)+CHR$(227)
40 R$=""
50 T=0
60 N=0
70 H=0
80 K$=INKEY$:FOR K=1 TO 10 :NEXT K
90 IF K$="" THEN 80
100 IF K$="=" THEN 400
110 IF (K$="-") AND (N=0) THEN 300
120 IF (K$="+") OR (K$="*") OR (K$="/"
) OR (K$="-") THEN 330
130 IF (T=1) AND (N=0) THEN CLS
140 GOSUB 160
150 GOTO 80
160 H=0
170 IF K$<>"." THEN 220
180 CURSOR N+3,3:PRINT CHR$(128)
190 R$=R$+"."
200 N=N-4*(N<36)
210 RETURN
220 FOR I=1 TO 5
230 S=2*VAL(MID$(B$,I+5*VAL(K$),1))-1
240 CURSOR N+3,I:PRINT MID$(C$,S,1)
250 CURSOR N+4,I:PRINT MID$(C$,S+1,1)
260 NEXT I
270 R$=R$+K$
280 N=N-4*(N<36)
290 RETURN
300 CURSOR 0,3:PRINT CHR$(155)
310 R$="-"+R$
320 GOTO 80
330 IF H=1 THEN 360
340 A$=R$
350 R$=""
360 O$=K$
370 N=0
380 T=1
385 K$=""
390 GOTO 80
400 CLS
410 IF O$<>"+" THEN 430
420 A$=STR$(VAL(A$)+VAL(R$))
430 IF O$<>"*" THEN 450
440 A$=STR$(VAL(A$)*VAL(R$))
450 IF O$<>"-" THEN 470
460 A$=STR$(VAL(A$)-VAL(R$))
470 IF O$<>"/" THEN 490
480 A$=STR$(VAL(A$)/VAL(R$))
490 N=0
500 FOR J=1 TO LEN(A$)-1
510 K$=MID$(A$,J+1,1)
520 GOSUB 160
530 NEXT J
540 N=0
550 T=1
560 H=1
570 GOTO 80
```

Smokey

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : solitaire.
Difficulité du jeu : moyenne.

LE JEU :
Une colonne de fumée noire apparaît sur l'écran. Le joueur, représenté par un petit bonhomme, devient visible sur la droite. Il doit dans un premier temps se positionner sur la colonne de fumée, puis à partir de là, il va automatiquement avaler la fumée, à condition qu'il réussisse à se maintenir sur sa trace.

LE PROGRAMME :
La ligne 8Ø fait apparaître la fumée au bas de l'écran ; la ligne 9Ø permet de changer (vers la droite ou vers la gauche) la direction de la colonne, et la ligne 11Ø fait monter la colonne.
Le joueur apparaît grâce à la ligne 13Ø, et son déplacement est effectué par les lignes 19Ø et 21Ø.
Les lignes 27Ø et 32Ø se chargent de l'affichage du résultat.

POUR JOUER :
On doit lancer l'exécution du programme par RUN.
La fumée commence à s'élever, tandis que le joueur, apparu sous forme d'un petit bonhomme dans un coin de l'écran, va se déplacer sur elle à l'aide des 4 flèches. Dès qu'il arrive sur la fumée, il va automatiquement commencer à l'avaler, et devra essayer de suivre la colonne sans dévier (grâce aux touches de déplacement droite et gauche). Il gagne s'il arrive en bas de l'écran avant que la fumée n'arrive en haut.
L'affichage final indique alors BRAVO ou PERDU.

EXTENSIONS POSSIBLES :
On pourra, tout en conservant l'idée de départ, imposer d'autres règles : par exemple avaler toute la fumée, qui pourra aussi repartir d'un autre endroit si l'on dévie de la colonne initiale.
On pourra aussi améliorer les contrôles de fin de partie, et la qualité du graphisme.

SMOKEY

```
10 CLS
20 PATTERN C# 128,"18180A7E48182424"
30 S=0
40 X=22
50 U=10
60 V=12
70 Y=INT(14*RND(1))+3
80 CURSOR Y,X:PRINT CHR$(229)
90 Y=Y+(RND(1)<.5)*(Y<17)-(RND(1)<.5)*
(Y>3)
100 T=1
110 X=X-T
120 IF X=1 THEN 270
130 CURSOR V,U:PRINT CHR$(128)
140 IF T=0 THEN 170
150 FOR PAUSE=1 TO 050
160 NEXT PAUSE
170 CURSOR V,U:PRINT " "
180 K$=INKEY$
190 U=U-(K$=CHR$(31))*(T<20)*(X<20)+(K
$=CHR$(30))*(T>0)*(X>1)
200 U=U+1-T
210 V=V+(K$=CHR$(28))*(V<17)-(K$=CHR$(
29))*(V>1)
220 COD=VPEEK(U*40+V+15362)
230 IF COD<>229 THEN 80
240 T=0
250 S=S+1
260 IF U<>20 THEN 130
270 CLS
280 IF T<>0 THEN 310
290 PRINT "BRAVO"
300 GOTO 320
310 PRINT "PERDU"
320 PRINT "SCORE : ";S
```

Course de voitures

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu de réflexe (1 joueur).
Difficulté du jeu : moyenne.

LE JEU :
Vous circulez sur une route très sinueuse, ne comportant aucun obstacle, et vous devez essayer de la suivre de plus longtemps possible. La trace de votre parcours reste affichée en permanence, et votre score est affiché dès que vous avez un accident.

LE PROGRAMME :
L'impression de la route est assurée par les lignes 11Ø à 13Ø.
La ligne 14Ø imprime la voiture.
La ligne 9Ø contrôle son mouvement, et la ligne 16Ø détermine si la route fait ou non des virages.
La ligne 2ØØ regarde si l'on a touché le bord de la route.
La ligne 26Ø incrémente le score, et la ligne 27Ø efface l'ancienne position de la voiture (une trace claire apparaît à l'emplacement de sa trajectoire).

POUR JOUER :
Lancer l'exécution du programme par l'ordre RUN, le dessin de la route commence à évoluer, et votre véhicule, représenté par un caractère redéfini, apparaît.
Pour le déplacer vers la droite ou vers la gauche, vous devrez utiliser les touches de déplacement ← et →. Il vous faudra éviter le plus longtemps possible de toucher une bordure noire, ce qui provoquera l'arrêt du programme et l'affichage de la mention ACCIDENT, suivie de l'indication sur le score réalisé.
On pourra arrêter le déroulement de la partie à tout moment en pressant la touche BREAK.

EXTENSIONS POSSIBLES :
On peut envisager de jouer à plusieurs, chacun son tour, en enregistrant le meilleur score.
On peut aussi envisager d'élargir la piste, et de faire courir plusieurs voitures.
On pourra aussi s'inspirer des jeux d'arcades qui sont particulièrement nombreux sur ce thème.

COURSE DE VOITURES

```
10 CLS
20 PATTERN C# 128,"D6FEC61010447C54"
30 A=9
40 B=A
50 X=A+1
60 Y=A-3
70 S=0
80 PRINT
90 K$=INKEY$
100 B=B-(K$=CHR$(28))+(K$=CHR$(29))
110 CURSOR Y,X:PRINT CHR$(229);CHR$(22
9)
120 CURSOR Y+2,X:PRINT "
130 CURSOR Y+5,X:PRINT CHR$(229);CHR$(
229)
140 CURSOR B,A:PRINT CHR$(128)
150 C=B
160 Y=Y+2*RND(1)*(Y>3)-2*RND(1)*(Y<16)

170 FOR PAUSE=1 TO 100
180 NEXT PAUSE
190 COD=VPEEK(40*(A+1)+B+15362)
200 IF COD<>229 THEN 260
210 CURSOR 2,20:PRINT "ACCIDENT","SCOR
E :";S
220 S=S-1
230 FOR PAUSE=1 TO200
240 NEXT PAUSE
250 CURSOR 2,20:FOR I=1 TO 30:PRINT "
";:NEXT I
260 S=S+1
270 CURSOR C,A:PRINT " "
280 GOTO 80
```

Echecs

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu de réflexion, pour deux joueurs.
Difficulté du jeu : difficile.

LE JEU :
Il n'était pas question en utilisant le Basic de réaliser un programme permettant au YENO de jouer aux échecs.
Le programme proposé permettra seulement à deux joueurs de disputer une partie en se servant de l'écran du téléviseur comme plateau de jeu. Il n'y a pas de contrôle de validité des coups, mais seulement gestion de la position des pièces sur l'échiquier. Les pièces sont représentées par leur initiale, les pions par P, les tours par T, les fous par F, les cavaliers par C, les dames par D et les rois par R. La couleur du fond permet de distinguer pièces noires et pièces blanches.

LE PROGRAMME :
La ligne 1Ø définit l'échiquier qui est affiché par la boucle I (16Ø-18Ø).
Les lignes 2Ø à 8Ø redéfinissent les caractères T, C, F, R, D, P (écrits blanc sur fond noir).
La boucle I (9Ø-11Ø) lit l'échiquier initial stocké en DATA.
La ligne 22Ø change de joueur. Les modifications des lignes de départ et d'arrivée de l'échiquier sont respectivement effectuées par les lignes 25Ø et 26Ø.
La ligne 23Ø détermine si la case de départ est noire ou blanche.

POUR JOUER :
On lance l'exécution par RUN.
L'échiquier est dessiné, et l'on voit apparaître la question JOUEUR BLANC L, C (DEPART) ; on introduit la ligne, puis la colonne de la pièce à manœuvrer, sous la forme, par exemple : 2,4 (pion blanc de la ligne 2 et de la colonne 4) suivi de CR.
On a ensuite L, C (ARRIV.) : on tape alors la ligne et la colonne de la case d'arrivée, sous la forme, par exemple : 4,4 suivi de CR.

EXTENSIONS POSSIBLES :
On pourra chercher à réaliser un graphisme plus précis (un exemple possible est d'ailleurs fourni dans ce même ouvrage). On pourra aussi sans trop de difficultés introduire le contrôle du mouvement des pièces. Il ne restera alors plus qu'à écrire le programme de jeu du YENO ! Bon courage !

```
10 DATA " 12345678","1TCFRDFCT","2PPPP
PPPP","3# # # # ","4 # # # #","5# # #
# ","6 # # # #","7pppppppp","8tcfrdfct
"
20 PATTERN C# 116, "FF83EFEFEFEFEFEF"
30 PATTERN C# 99, "FFC7BBBFBFBFBBC7"
40 PATTERN C# 102, "FF83BFBF87BFBFBF"
50 PATTERN C# 100, "FF87DBDBDBDBDB87"
60 PATTERN C# 114, "FF87BBBB87AFB7BB"
70 PATTERN C# 112, "FF87BBBB87BFBFBF"
80 PATTERN C# 35, "FFFFFFFFFFFFFFFF"
90 FOR I=0 TO 8
100 READ L$(I)
110 NEXT I
120 C$(0)="BLANC"
130 C$(1)="NOIR"
140 H=0
150 CLS
160 FOR I=0 TO 8
170 CURSOR 15,5+I:PRINT L$(I)
180 NEXT I
190 PRINT :PRINT :PRINT :PRINT "JOUEUR
 ";C$(H)
200 INPUT "L,C (DEPART) :";LD,CD
210 INPUT "L,C (ARRIV.) :";LA,CA
220 H=ABS(1-H)
230 COD=-32*((LD+CD)/2<>INT(LD+CD)/2)-
35*((LD+CD)/2=INT((CD+LD)/2))
240 S$=MID$(L$(LD),CD+1,1)
250 L$(LD)=LEFT$(L$(LD),CD)+CHR$(COD)+
RIGHT$(L$(LD),8-CD)
260 L$(LA)=LEFT$(L$(LA),CA)+S$+RIGHT$(
L$(LA),8-CA)
270 GOTO 150
```

Danger spatial

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu de réflexe, pour un joueur.
Difficulté du jeu : moyenne.

LE JEU :
Vous êtes aux commandes d'un engin spatial qui vient de pénétrer dans un nuage de météorites. Vous devez essayer de franchir le nuage sans heurter une seule météorite. Votre score sera affiché en fin de partie, lorsque vous entrez en collision avec une météorite.

LE PROGRAMME :
La position de chaque météorite est déterminée par la ligne 4Ø ; elle est imprimée par la ligne 6Ø.
L'avance des météorites est réalisée par une instruction PRINT ligne 8Ø.
Les lignes 9Ø et 12Ø vérifient si vous avez ou non heurté une météorite.
La ligne 14Ø dessine votre vaisseau spatial, et la ligne 16Ø permet de le déplacer.
Le compteur se trouve en ligne 17Ø.

POUR JOUER :
Le jeu est lancé par l'ordre RUN.
Votre vaisseau apparaît vers le milieu de l'écran, et les météorites viennent à votre rencontre en partant du bas de l'écran. Pour vous déplacer, vous pouvez utiliser les touches de déplacement horizontal (← et →).
On peut arrêter la partie à tout moment en tapant BREAK. Sinon, le jeu s'arrête lorsque vous heurtez une météorite, et le score est alors affiché.

EXTENSIONS POSSIBLES :
Il serait intéressant de rendre le jeu de plus en plus difficile en faisant apparaître de plus en plus de météorites au fur et à mesure du déroulement de la partie.
On peut aussi essayer de détruire les météorites en les pulvérisant au fur et à mesure de leur apparition.

```
10 CLS
20 X=12
30 Z=0
40 H=INT(30*RND(1))+3
50 IF RND(1)>.7 THEN 70
60 CURSOR H,23:PRINT "Z"
70 IF RND(1)>.6 THEN 100
80 CURSOR 3+INT(30*RND(1)),20:PRINT "Z
"
90 C0=VPEEK(40*11+X-1+15362)
100 C1=VPEEK(40*11+X+1+15362)
110 C2=VPEEK(40*11+X+1+15362)
120 IF (C0=90) OR (C1=90) OR (C2=90)
THEN 190
130 CURSOR X-3,10:PRINT "       "
140 CURSOR X,11:PRINT "H"
150 K$=INKEY$
160 X=X+(K$=CHR$(28))*(X<32)-(K$=CHR$(
29))*(X>3)
170 Z=Z+1
180 GOTO 40
190 CLS
200 CURSOR 20,0:PRINT "SCORE :";Z
```

Tic tac toe

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion (deux joueurs).
Difficulté du jeu : facile.

LE JEU :
Un grand classique des jeux d'écoliers, proposé ici dans une version ne comportant pas le jeu machine ; chaque joueur devra placer à son tour un Ø ou un X dans une case libre de la grille de jeu, et le premier qui réussit un alignement de 3 symboles identiques a gagné (l'alignement peut être horizontal, vertical, ou suivant l'une des deux diagonales).

LE PROGRAMME :
Le terrain de jeu est dessiné à l'aide de deux boucles I (3Ø-8Ø et 9Ø-14Ø).
La ligne 16Ø s'occupe d'afficher le symbole X ou Ø correspondant au joueur qui a le trait.
Un symbole clignotant (ligne 23Ø) est affiché en haut et à gauche du terrain, même si la case est déjà occupée.
Les lignes 26Ø et 27Ø permettent de déplacer le symbole sur le terrain de jeu.
La ligne 28Ø vérifie si la case où l'on veut jouer est libre, et si l'on désire y jouer.
La ligne 29Ø affiche le coup, et la ligne 33Ø réalise le changement de joueur.

POUR JOUER :
Le terrain de jeu sera affiché dès la frappe de RUN.
Un symbole clignotant apparaît alors en haut et à gauche de l'écran ; le joueur peut le déplacer sur le terrain de jeu à l'aide des 4 flèches.
Lorsqu'il arrive sur la case où il désire jouer, il suffit qu'il appuie sur la touche Ø, le coup sera refusé si la case n'est pas libre.
Si la case était libre, le symbole qui correspond à ce joueur sera inscrit, et on change de joueur.
Lorsque la partie est finie, il convient de taper BREAK pour sortir du programme. On pourra ensuite faire une autre partie avec RUN.

EXTENSIONS POSSIBLES :
On pourrait réaliser après chaque coup un contrôle pour vérifier si l'un des joueurs a gagné. On peut aussi réaliser un jeu machine mais il faudrait éviter qu'il soit par trop systématique (par exemple, la machine jouant toujours son premier coup sur la case centrale).

TIC TAC TOE

```
10 CLS
20 PRINT TAB(10);"JOUEUR ":PRINT :PRIN
T :PRINT
30 FOR I=5 TO 9
40 CURSOR I,6:PRINT CHR$(144)
50 CURSOR I,8:PRINT CHR$(144)
60 CURSOR 6,I:PRINT CHR$(144)
70 CURSOR 8,I:PRINT CHR$(144)
80 NEXT I
90 FOR I=4 TO 10
100 CURSOR I,4:PRINT CHR$(229)
110 CURSOR I,10:PRINT CHR$(229)
120 CURSOR 4,I:PRINT CHR$(229)
130 CURSOR 10,I:PRINT CHR$(229)
140 NEXT I
150 K=0
160 IF K<>0 THEN 190
170 CURSOR 19,0:PRINT "X"
180 GOTO 200
190 CURSOR 19,0:PRINT "0"
200 I=1
210 J=1
220 C=VPEEK(40*(2*I+3)+2*J+3+15362)
230 CURSOR 2*J+3,2*I+3:PRINT "*"
240 CURSOR 2*J+3,2*I+3:PRINT CHR$(C)
250 K$=INKEY$
260 I=I+(K$=CHR$(31))*(I<3)-(K$=CHR$(3
0))*(I>1)
270 J=J+(K$=CHR$(28))*(J<3)-(K$=CHR$(2
9))*(J>1)
280 IF (C<>32) OR(K$<>"0") THEN 220
290 IF K=0 THEN 320
300 CURSOR 2*J+3,2*I+3:PRINT "0"
310 GOTO 330
320 CURSOR 2*J+3,2*I+3:PRINT "X"
330 K=ABS(K-1)
340 GOTO 160
```

Gouttes

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : dessin.
Difficulté du jeu : aucune.

LE JEU :
C'est un véritable programme de dessin animé que l'on peut considérer, en fait, comme un interlude ; un récipient est dessiné sur l'écran, vu en coupe, et il contient déjà des obstacles. Une goutte d'eau, symbolisée par un carré gris, apparaît au-dessus du récipient et tombe en remplissant le récipient au fur et à mesure. Sa trajectoire, lorsqu'elle rencontre un obstacle, est entièrement déterminée par le hasard : elle part à droite ou à gauche, jusqu'à ce qu'il n'y ait plus aucune possibilité. Alors, une autre goutte tombe, et ainsi de suite, jusqu'au débordement final.

LE PROGRAMME :
Les lignes 7Ø à 9Ø dessinent le bassin et la boucle (1ØØ-12Ø), les obstacles.
La série de gouttes est dessinée en ligne 21Ø.
Le sous-programme (47Ø-57Ø) regarde si la voie est libre ou si un obstacle se trouve sur la trajectoire de la série de gouttes.
Les diverses positions de toutes les gouttes composant la série sont stockées dans deux tableaux L et C périodiquement remis à jour à chaque déplacement.
En changeant la ligne 6Ø, on peut modifier le nombre de gouttes constituant une série.

POUR JOUER :
Le RUN déclenche le dessin du récipient et des obstacles, une série de gouttes apparaît, se met à tomber et finit par se trouver une place. Une autre série apparaît et l'on recommence.
Le programme s'arrête de lui-même lorsque le récipient déborde ; on peut aussi l'arrêter à tout moment en tapant BREAK.

EXTENSIONS POSSIBLES :
On peut, bien entendu, ajouter la couleur et le son et aussi envisager un jeu dans lequel un petit personnage devrait échapper aux gouttes ou à la noyade.

GOUTTES

```
10 CLS
20 PATTERN C# 128, "1818181818181818"
30 PATTERN C# 129, "FFFF000000000000"
40 PATTERN C# 131, "00183C7E7E3C1800"
50 ROC=80
60 LNG=6
70 CURSOR 3,23:FOR I=1 TO 30:PRINT CHR
$(129);:NEXT I
80 FOR I=1 TO 22:CURSOR 3,I:PRINT CHR$
(128):NEXT I
90 FOR I=1 TO 22:CURSOR 32,I:PRINT CHR
$(128):NEXT I
100 FOR I=1 TO ROC
110 CURSOR 4+25*RND(1),2+21*RND(1):PRI
NT CHR$(229)
120 NEXT  I
130 FOR I=3 TO 31
140 CK=VPEEK(40*1+I+15362)
150 IF CK=32 THEN 180
160 NEXT  I
170 END
180 CC=4+27*RND(1)
190 IF LL=131 THEN 180
200 FOR I=1 TO LNG:CURSOR CC,I:PRINT C
HR$(131):NEXT I
210 FOR I=1 TO LNG
220 C(I)=CC
230 L(I)=I
240 NEXT I
250 U=L(LNG)+1
260 V=C(LNG)
270 GOSUB 460
280 IF COD=32 THEN 250
290 V=C(LNG)+1
300 U=L(LNG)+1
310 GOSUB 460
320 IF COD=32 THEN 250
330 U=L(LNG)+1
340 V=C(LNG)-1
350 GOSUB 460
360 IF COD=32 THEN 250
370 U=L(LNG)
380 V=C(LNG)-1
390 GOSUB 460
400 IF COD=32 THEN 250
410 U=L(LNG)
420 V=C(LNG)+1
430 GOSUB 460
440 IF COD=32 THEN 250
450 GOTO 130
460 COD=VPEEK(40*U+V+15362)
470 IF COD<>32 THEN 560
480 CURSOR V,U:PRINT CHR$(131)
490 CURSOR C(1),L(1):PRINT " "
500 FOR I=1 TO LNG-1
510 C(I)=C(I+1)
520 L(I)=L(I+1)
530 NEXT I
540 L(LNG)=U
550 C(LNG)=V
560 RETURN
```

Obstacles

REFERENCES :
Difficulté de programmation : assez difficile.
Catégorie de jeu : jeu de réflexe (pour un joueur).
Difficulté du jeu : moyenne.

LE JEU :
Des obstacles se déplacent sur l'écran pour venir à la rencontre d'une petite voiture symbolisée par un 8 qui dispose de la possibilité de se déplacer vers la gauche ou vers la droite pour les éviter et qui devra tenir le plus longtemps possible.
La distance parcourue sera affichée en permanence, et le jeu se termine lorsque la voiture rencontre, soit le bord du terrain, soit un obstacle.

LE PROGRAMME :
La voiture du joueur est imprimée grâce à la ligne 6Ø.
Les lignes 7Ø, 8Ø, 11Ø fabriquent la route (les lignes 8Ø et 1ØØ bien qu'identiques sont toutes deux indispensables, car la valeur de RND ne sera, en principe, pas la même sur ces deux lignes).
La ligne 14Ø affiche la route et le score.
La ligne 17Ø permettra de déplacer le véhicule.
Les lignes 18Ø, 19Ø et 22Ø permettent de contrôler s'il y a ou non un obstacle et arrêtent le programme s'il y a lieu.
Le score est incrémenté par la ligne 2ØØ.

POUR JOUER :
On lance le programme en tapant RUN.
La route et les obstacles apparaissent en bas de l'écran et montent vers la voiture représentée par un 8. La voiture peut se déplacer à l'aide des flèches de déplacement horizontal.
La rencontre avec un obstacle interrompt le déroulement du programme et provoque l'affichage du score.
On peut, cependant, sortir du programme à tout moment en utilisant la touche BREAK.

EXTENSIONS POSSIBLES :
Il sera assez facile d'améliorer le dessin de la voiture, de la route, des obstacles. On pourra aussi introduire la couleur, admettre un certain nombre d'accidents, enregistrer le meilleur score (pour plusieurs joueurs), éventuellement dessiner plusieurs voitures, etc.

OBSTACLES

```
10 CLS
20 Y=7
30 S=-12
40 CLS
50 PATTERN C# 124, "1818181818181818"
60 CURSOR Y,10:PRINT "8"
70 A$="¦       ¦"
80 R=7*RND(1)+2
90 A$=LEFT$(A$,R-1)+CHR$(229)+RIGHT$(A
$,9-R)
100 R=7*RND(1)+2
110 A$=LEFT$(A$,R-1)+CHR$(229)+RIGHT$(
A$,9-R)
120 CURSOR 20,22:PRINT "         "
130 R=0:IF S>0 THEN R=S
140 PRINT A$;TAB(23);R
150 CURSOR 0,8:PRINT "         "
160 K$=INKEY$
170 Y=Y+(K$=CHR$(29))-(K$=CHR$(28))
180 C=VPEEK(40*10+Y+15362)
190 IF (C=229) OR (C=124) THEN 220
200 S=S+1
210 GOTO 60
220 CURSOR 15,0:PRINT "ACCIDENT !!!"
230 END
```

Captures dans l'espace

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : réflexe (1 seul joueur).
Difficulté du jeu : moyenne.

LE JEU :
Votre vaisseau spatial navigue dans un dangereux champ de météorites et votre mission est de vous emparer de la plus grande quantité possible de minérai d'urane, un métal indispensable que l'on ne peut trouver que dans ces lieux particulièrement dangereux.
Vous devez arriver à manœuvrer pour recueillir l'urane, sans vous faire heurter par une météorite.

LE PROGRAMME :
La ligne 30 permet de faire apparaître les météorites.
La ligne 50 fera apparaître le minerai d'urane.
Leur déplacement, sur l'écran, est assuré par un PRINT ligne 60.
Les lignes 70 et 90 contrôlent le déroulement du jeu (a-t-on heurté une météorite ou a-t-on capturé un bloc d'urane).
La rencontre d'une météorite provoque la fin de partie.
Votre vaisseau spatial est imprimé par la ligne 120 et son déplacement est contrôlé par la ligne 140.

POUR JOUER :
On lance le programme par RUN et les météorites commencent à monter sur l'écran, ainsi que les blocs d'urane.
Les touches de déplacement horizontal (← et →) seront utilisées pour déplacer votre vaisseau et récupérer l'urane, mais le moindre heurt avec une météorite provoquera la fin de la partie et l'affichage de quantité d'urane capturée.
On pourra aussi arrêter à tout moment en tapant BREAK.

EXTENSIONS POSSIBLES :
On peut rendre le jeu de plus en plus difficile en augmentant progressivement le nombre de météorites, ou encore, en minutant la durée de la partie.
On peut aussi prévoir un affichage permanent du score, introduire un deuxième vaisseau spatial manœuvré par une puissance étrangère et qui cherche aussi à s'emparer de l'urane...

```
10 X=12
20 Z=0
30 CURSOR INT(30*RND(1))+2,20:PRINT CH
R$(229)
40 IF RND(1)>.5 THEN 60
50 CURSOR INT(30*RND(1))+2,20:PRINT "0
"
60 PRINT :PRINT :PRINT
70 C=VPEEK(40*12+X+15362)
80 IF C=229 THEN 160
90 IF C<>48 THEN 110
100 Z=Z+1
110 CURSOR X-2,10:PRINT "       "
120 CURSOR X,12:PRINT "V"
130 K$=INKEY$
140 X=X+(K$=CHR$(28))*(X<32)-(K$=CHR$(
29))*(X>1)
150 GOTO 30
160 CLS
170 PRINT "SCORE ";Z
```

Plusieurs en un

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : deux joueurs.
Difficulté du jeu : variable, selon l'utilisation.

LE JEU :
Plusieurs jeux sont en réalité possibles avec ce programme qui va se comporter comme un crayon électronique vous permettant d'inscrire des symboles O ou X, en n'importe quel emplacement de votre feuille-écran de télévision. On pourra, par exemple, l'utiliser pour un jeu de morpions (le premier à réaliser un alignement de 5 pions identiques gagne) ou pour un GO (on réalise des encadrements des pions adverses) ou tout autre jeu, selon son imagination.

LE PROGRAMME :
Le terrain de jeu est délimité par 4 symboles redéfinis et mis en place par les lignes 4Ø à 11Ø.
Le changement de joueur est effectué par la ligne 13Ø.
Le joueur qui a le trait voit le symbole qu'il a choisi (O ou X) se placer en limite du cadre (lignes 14Ø et 15Ø).
Les lignes 2ØØ et 21Ø permettent aux deux joueurs de déplacer le symbole sur le terrain.
Les lignes 23Ø et 24Ø contrôlent si le déplacement est possible (case libre et non dépassement des limites).
La ligne 29Ø permet d'écrire sur une case libre.

POUR JOUER :
RUN provoque l'affichage des limites du terrain et du numéro du joueur, ainsi que des symboles correspondants (O ou X) que l'on devra déplacer sur l'écran à l'aide des 4 touches de direction.
Il n'est pas possible de venir sur une case déjà occupée et il faudra appuyer sur la touche Ø pour inscrire son propre symbole dans une case disponible.
Dès que cela a été fait, le symbole du second joueur apparaît dans la colonne de gauche, à peu près au milieu, et le joueur procédera de la même manière que son adversaire.
Pour sortir du programme, il suffira de taper BREAK.

EXTENSIONS POSSIBLES :
On pourra utiliser l'écran tout entier comme terrain de jeu et réaliser des contrôles en fonction des différents jeux que l'on désire pratiquer : contrôle d'alignements de 5 pièces pour le morpion, contrôle du nombre de pièces capturées pour le GO, etc.

```
10 CLS
20 D(0)=48
30 D(1)=88
40 PATTERN C# 128, "FFFFFFFFF0F0F0F0"
50 PATTERN C# 129, "FFFFFFFF0F0F0F0F"
60 PATTERN C# 130, "0F0F0F0FFFFFFFFF"
70 PATTERN C# 131, "F0F0F0F0FFFFFFFF"
80 CURSOR 3,1:PRINT CHR$(128)
90 CURSOR 32,1:PRINT CHR$(129)
100 CURSOR 32,22:PRINT CHR$(130)
110 CURSOR 3,22:PRINT CHR$(131)
120 J=0
130 J=ABS(J-1)
140 X=12
150 Y=3
160 CURSOR Y,X:PRINT CHR$(D(J))
170 K$=INKEY$
180 A=X
190 B=Y
200 X=X+(K$=CHR$(31))*(X<22)-(K$=CHR$(
30))*(X>1)
210 Y=Y+(K$=CHR$(28))*(Y<32)-(K$=CHR$(
29))*(Y>3)
220 CURSOR B,A:PRINT " "
230 C=VPEEK(40*X+Y+15362)
240 IF (C<>48) AND (C<>88) THEN 280
250 CURSOR Y,X:PRINT CHR$(C)
260 X=A
270 Y=B
280 IF K$<>"0" THEN 160
290 CURSOR Y,X:PRINT CHR$(D(J))
300 GOTO 130
```

Niveau 5

Nom du programme	*Thèmes et particularités*
Horloge 2	Utilisation du générateur de caractères incorporé.
Dames	Pour deux joueurs : classique.
Anagrammes	Un bon jeu de réflexes, pour 2 joueurs.
Œufs	Ne laissez pas faire la pondeuse diabolique ! (1 joueur).
Affichage tournant (vertical)	Une possibilité à exploiter de votre YENO.
Quatre à quatre	Un jeu de réflexion un peu difficile, pour 2 joueurs.
Taquin de lettres	Un proverbe est caché dedans ; à vous de le découvrir.
Cibles	Un peu d'adresse, beaucoup de chance. Trés visuel.
Affichage publicitaire (2)	Un affichage de lettres géantes !
Anomalie	Exercez votre perspicacité dans la reconnaissance des symboles.
Othello	Pour deux joueurs ; il retournera les pions à votre place.

Présentation de certaines fonctions du Basic étendu

Les programmes présentés dans les quatre premières parties sont exécutables en Basic simple.

Le Basic graphique offre des possibilités, au niveau des dessins, largement accrues. Les programmes qui suivent utilisent cette puissante caractéristique du YENO.

SPRITES

La caractéristique la plus impressionnante de ce Basic est la faculté de créer et de gérer de petits lutins animés qui seront ensuite manipulés par le processeur vidéo. Nous ne reviendrons pas sur l'importante palette d'instructions y donnant accès (PATTERN et MAG).

Horloge 2

REFERENCES :
Basic Graphique.
Difficulté de programmation : difficile.
Catégorie de jeu : dessin.
Difficulté du jeu : aucune.

LE JEU :
Ce deuxième programme de simulation d'une horloge digitale grand format présente la particularité intéressante d'utiliser les lutins (SPRITES) du BASIC Graphique. Ce sont donc les caractères d'origine qui seront reproduits agrandis par l'instruction MAG 2.

LE PROGRAMME :
La ligne 12Ø permet l'introduction de l'heure sous la forme de 3 entiers de 2 chiffres.
La ligne 14Ø décode cet entier.
L'heure est affichée au moyen de SPRITES par la ligne 16Ø.

POUR JOUER :
Faire RUN, puis introduire 3 entiers de 2 chiffres (03:45:15 représente 3 heures 45 minutes et 15 secondes).
Attendre le quatrième top pour taper CR et l'affichage commence.

EXTENSIONS POSSIBLES :
On peut faire clignoter les deux points, ajouter une possibilité de réveil, prévoir un agenda automatique, faire apparaître un coucou chantant au moment des heures, etc.

HORLOGE 2

```
10 PATTERN S# 0,"3048484848483000"
20 PATTERN S# 1,"2060202020207000"
30 PATTERN S# 3,"708808102040F800"
40 PATTERN S# 3,"7088083008887000"
50 PATTERN S# 4,"30509090F8101000"
60 PATTERN S# 5,"F880F00808887000"
70 PATTERN S# 6,"708080F088887000"
80 PATTERN S# 7,"F888081020202000"
90 PATTERN S# 8,"7088887088887000"
100 PATTERN S# 9,"7088887808106000"
110 CLS
120 INPUT "HEURE (HH:MM:SS) :";H$:TIME
$=H$
130 SCREEN 2,2
140 H$=LEFT$(TIME$,2)+MID$(TIME$,4,2)
150 FOR I=1 TO 4:S(I)=VAL(MID$(H$,I,1)
):NEXT I
160 MAG 2:FOR I=1 TO 4:SPRITE I,(75+20
*I,75),S(I),1:NEXT I
170 GOTO 140
```

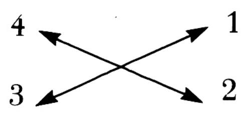
Dames

REFERENCES :
Basic Graphique.
Difficulté de programmation : très difficile.
Catégorie de jeu : deux joueurs.
Difficulté du jeu : difficile.

LE JEU :
C'est le jeu classique de dames à la française, mais un peu incomplet puisqu'il ne prévoit pas la possibilité de faire des dames. La règle du jeu reste classique, mais les contrôles ont été un peu simplifiés. Les mouvements des pièces sont entrés en une seule manœuvre (position de départ et direction du déplacement) et les prises sont réalisées automatiquement, si elles sont possibles.
Par contre le changement de joueur étant réalisé à chaque coup, il faudra éventuellement taper Ø pour pouvoir rejouer s'il y avait eu saut ou si le coup n'était pas possible.

LE PROGRAMME :
La boucle I ligne 5Ø dessine le damier (1Ø cases sur 1Ø cases) et les pions (X ou Ø).
La ligne 8Ø indique qui a le trait.
Le changement de joueur est traité par la ligne 8Ø (entrée d'un Ø) également.
La ligne 9Ø calcule le déplacement, qui sera réalisé lorsqu'il est possible, par la boucle I (1ØØ-15Ø).
La ligne 1ØØ calcule la case d'arrivée (elle tient compte du cas où l'on arriverait en ligne Ø).
La ligne 11Ø teste les cases à parcourir. La ligne 13Ø traite le cas d'un déplacement et, 14Ø celui d'un saut. La ligne 14Ø effectue les éventuelles modifications, et la ligne 16Ø réalise le changement de joueur.

POUR JOUER :
Dès le lancement du programme par RUN, le damier est affiché avec ses pions, et un caractère O inversé apparaît en dessous pour indiquer que le trait appartient au joueur qui a choisi le O.
On joue en indiquant par un mot de 3 chiffres (pouvant commencer par un Ø) la ligne et la colonne de la pièce à déplacer, puis la direction du déplacement (codée de 1 à 4) suivant la convention :



Par exemple 931 signifie que le O situé en ligne 9, colonne 3, se déplace dans la direction 1 (ne pas oublier de faire CR).
De même Ø63 signifiera pièce X, ligne Ø, colonne 6, direction 3.
Le changement de joueur étant effectué automatiquement, il faudra taper Ø pour rejouer si le coup est refusé ou s'il y a eu prise (toujours terminer par CR). Le programme ne contrôle pas l'arrivée à dame, et n'interdit pas les déplacements en arrière.

EXTENSIONS POSSIBLES :
Il sera assez facile de commencer par améliorer les contrôles (changement de joueur, coup en arrière qui ne correspondrait pas à une prise, sortie du terrain). On pourra envisager la création de dames, leur déplacement, et éventuellement écrire un programme permettant au YENO de se mêler avec de bonnes chances de réussite à vos passionnantes parties.

DAMES

```
10 CLS
20 PATTERN C# 35, "FFFFFFFFFFFFFFFF"
30 PATTERN C# 111, "F88888888888F800"
40 PATTERN C# 120, "8888502050888800"
50 FOR I=0 TO 9:CURSOR 12+I,3:PRINTMID
$(STR$(I),2,1):CURSOR 9,I+5:PRINTI;TAB
(12-(I/2<>INT(I/2)));"o o # x x":NEXT
I
60 H=1
70 A$="x":IF H<>0 THEN A$="o"
80 BEEP:CURSOR 5,20:PRINT "JOUEUR ";A$
:CURSOR 14,20:INPUT R$:IF R$="0" THEN
170
90 D=VAL(MID$(R$,3,1)):D=9*((D=1)-(D=3
))+11*((D=4)-(D=2)):B$=R$:T=VAL(MID$(R
$,1,2)):R=2
100 FOR I=1 TO 3:R$=STR$(ABS(T+(I-1)*D
)):IF LEN(R$)<2 THEN R$="0"+R$
110 X=6+VAL(LEFT$(R$,1)):Y=14+VAL(MID$
(R$,2,1)):COD=VPEEK(40*X+Y+2+15362):IF
 R=2 THEN B$=MID$(B$,1,I-1)+CHR$(COD)+
MID$(B$,I+1,3)
120 PRINTX;Y
130 AA$=A$+"#":IF MID$(B$,1,2)=AA$ THE
N B$="#"+A$+MID$(B$,3,1)
140 IF MID$(B$,1,1)=A$ AND MID$(B$,2,1
)<>A$ AND MID$(B$,1,1)="#" THEN B$="##
"+A$
150 CURSOR Y,X:PRINTMID$(B$,I,1):NEXT
I
160 R=R-1:IF R<>0 THEN 100
170 H=1-H:GOTO 70
```

Anagrammes

REFERENCES :
Basic Graphique.
Difficulté de programmation : difficile.
Catégorie de jeu : jeu de réflexe (2 joueurs).
Difficulté du jeu : selon le niveau.

LE JEU :
Des lettres sont choisies au hasard par le YENO, le nombre de lettres étant fonction du niveau de difficulté désiré. Puis les lettres du mot ainsi obtenu sont brouillées, et affichées au bas de l'écran ; un nouveau mot est alors confectionné, de façon aléatoire, à partir des mêmes lettres et il est affiché sous le premier, et ainsi de suite ; les deux joueurs (gauche et droit) disposent chacun d'une touche (1 ou Ø) et doivent l'appuyer dès qu'ils voient s'afficher au bas de l'écran un mot déjà présent sur l'écran ; leur score est alors évalué, et un message correspondant à leur résultat est délivré.

LE PROGRAMME :
Le niveau du jeu est entré à la ligne 1Ø. La boucle I (3Ø) réalise le choix des lettres.
Une deuxième boucle I (4Ø-1ØØ) effectue le brouillage, et la ligne 6Ø permet de détecter la pression d'une touche.
La ligne 11Ø affiche le nouveau mot, et fait monter d'une ligne sur l'écran les mots déjà obtenus.
Les lignes 12Ø et 13Ø cherchent quel est celui des deux joueurs qui a pressé une touche.
La boucle I (13Ø-16Ø) explore les mots présents sur l'écran ligne par ligne, alors que la boucle J (13Ø-14Ø) saisit un par un les caractères de ces mots.
On obtient ainsi un nouveau mot qui est comparé au mot initial en ligne 15Ø et le résultat BRAVO apparaît si c'est exact.
Sinon, la partie s'arrête sur l'affichage PERDU (ligne 17Ø).

POUR JOUER :
On lance le programme par RUN.
On devra ensuite entrer le niveau de difficulté (en principe de 1 à 3) et faire CR. Les mots s'affichent en partant du bas de l'écran, et montent au fur et à mesure ; si l'un des joueurs croit reconnaître dans le mot qui apparaît un des mots déjà affichés, il devra immédiatement presser sa touche (1 ou Ø), et son résultat sera affiché après un court instant. La partie sera alors terminée.

EXTENSIONS POSSIBLES :
On pourra utiliser d'autres affichages que des mots, modifier les temps d'affichage, faire participer plus de deux joueurs, introduire le son et la couleur, etc.

ANAGRAMMES

```
10 CLS:INPUT "NIVEAU (1/3) :";N
20 CLS:S$="":FOR I=1 TO 3+N:S$=S$+CHR$
(INT(65+26*RND(1))):NEXT I
30 A$="":B$=S$
40 FOR I=1 TO LEN(S$)
50 N=INT(LEN(S$)*RND(1)+1)
60 K$=INKEY$:IF (K$="1") OR (K$="0") T
HEN 120
70 IF MID$(B$,N,1)="-" THEN 50
80 A$=A$+MID$(B$,N,1)
90 B$=LEFT$(B$,N-1)+"-"+RIGHT$(B$,LEN(
S$)-N)
100 NEXT I
110 CURSOR 9,23:PRINTA$:S$=A$:GOTO 30
120 H$="DROIT":IF K$="1" THEN H$="GAUC
HE"
130 FOR I=1 TO 21:B$="":FOR J=1 TO LEN
 (S$)
140 CC=VPEEK(40*I+8+J+15362):B$=B$+CHR
$(CC):NEXT J
150 IF B$=S$ THEN CURSOR LEN(S$)+11,I:
PRINT"<== BRAVO ";H$:END
160 NEXT I
170 PRINT TAB(LEN(S$)+11);"PERDU ";H$
```

Œufs

REFERENCES :
Basic Graphique.
Difficulté de programmation : difficile.
Catégorie de jeu : solitaire.
Difficulté du jeu : moyenne.

LE JEU :
Un oiseau mystérieux se déplace en battant des ailes au hasard sur l'écran. Il pond des œufs (représentés par des O) que vous devez essayer de récupérer. Votre score (nombre d'œufs récupérés) sera affiché en permanence, et la partie se déroule en un laps de temps défini.

LE PROGRAMME :
L'oiseau est dessiné en deux parties (pour le mouvement) aux lignes 2Ø et 3Ø ; son déplacement aléatoire est géré par le sous-programme 6Ø-13Ø (en particulier les lignes 11Ø et 12Ø).
La ligne 3Ø permet de reboucler le programme sur la ligne 2Ø.
Le déplacement du joueur est régi par la ligne 4Ø, et le test permettant de détecter la présence d'un œuf est effectué à la ligne 5Ø.
La ligne 5Ø permet l'affichage du score.
La fin de jeu est détectée en ligne 6Ø.

POUR JOUER :
On lancera le programme avec RUN.
L'oiseau commence alors à se déplacer au hasard sur l'écran, et se met à pondre des œufs. Lorsque le symbole qui représente le joueur (caractère redéfini) apparaît sur l'écran, celui-ci peut se déplacer en utilisant les 4 touches de déplacement. Il vous faut dès lors essayer de récupérer un maximum d'œufs avant la fin de la partie.
On pourra arrêter de jouer à tout moment en tapant BREAK.

EXTENSIONS POSSIBLES :
Sur le même principe, on pourra facilement imaginer d'autres thèmes de jeu. On pourra bien sûr introduire la couleur, les sons, améliorer le graphisme, et éventuellement, définir un jeu à plusieurs joueurs, chacun essayant de récupérer un maximum d'œufs.

OEUFS

```
10 CLS:P=200:X INT(20*RND(1)):Y=INT(20
*RND(1)):R=0:S=0:I=0
20 CURSOR X,Y:PRINT CHR$(147);CHR$(148
):K$=INKEY$:IF K$<>"" THEN GOSUB 40
30 GOSUB 60:CURSOR X,Y:PRINT CHR$(148)
;CHR$(147):GOSUB 60:GOTO 20
40 CURSOR S,R:PRINT " ":R=R+(K$=CHR$(3
1))*(R<20)-(K$=CHR$(30))*(R>0):S=S+(K$
=CHR$(28))*(S<30)-(K$=CHR$(29))*(S>0)
50 CC=VPEEK(40*R+S+15362):E=(CC=79):T=
T-E:CURSOR 11,21:PRINT "SCORE :";T: RE
TURN
60 A=X:B=Y:P=P-1:IF P=0 THEN PRINT "C'
EST FINI.....":END
70 CURSOR S,R:PRINT "H"
80 FOR U=1 TO 50:NEXT U
90 CURSOR A,B:PRINT "  "
100 CURSOR A,B:PRINT "O"
110 X=ABS(X+4*RND(1)-2):Y=ABS(Y+2*RND(
1)-1):IF X>29 THEN X=1
120 IF Y>19 THEN Y=1
130 RETURN
```

Affichage tournant (vertical)

REFERENCES :
Basic Etendu.
Difficulté de programmation : facile.
Catégorie de jeu : affichage publicitaire.
Difficulté du jeu : facile.

LE JEU :
Le programme proposé réalise le défilement d'un texte, affiché en gros caractères, sur l'écran. L'utilisation des SPRITES a permis de simplifier considérablement le programme.

LE PROGRAMME :
La boucle I ligne 9Ø réalise l'affichage lettre après lettre en créant pour chacune d'entre-elles une SPRITE défilant verticalement. Les SPRITES étant gérées par le processeur vidéo, il suffit ensuite de faire une boucle infinie afin de ne pas les détruire.

POUR JOUER :
On doit d'abord lancer le programme par l'instruction RUN, puis on entre le texte, on tape ensuite CR et on regarde.
Le programme boucle sur lui-même et le défilement ne pourra être interrompu que par un appui sur la touche BREAK.

EXTENSIONS POSSIBLES :
Il suffira d'avoir un peu d'imagination pour adapter ce programme à toute sorte d'affichage de types publicitaires (il s'agit alors d'attirer l'attention, d'une part, et d'informer d'autre part). On pourra à cet effet utiliser la couleur, personnaliser les messages, utiliser plusieurs lignes, etc.

N.B. : Pour étendre ce programme à d'autres affichages, il sera nécessaire de redéfinir, par un PATTERN, les sprites d'autres caractères. Les "descripteurs" (suite hexadécimale codant le caractère pour l'instruction PATTERN) sont récupérables en mémoire Vidéo par un VPEEK. La table de générateur des caractères débute à la mémoire 6144 (H$ 18ØØ) pour "!" (ASCII = 33) : son descripteur se trouve sur les mémoires de 6144 à 6151 ; les caractères se suivent ensuite dans l'ordre de leur code ASCII.

AFFICHAGE TOURNANT (VERTICAL)

```
10 PATTERN S# 0, "8888502020202000"
20 PATTERN S# 1, "F88080F08080F800"
30 PATTERN S# 2, "88C8A89888888800"
40 PATTERN S# 3, "F888888888888F800"
50 CLS
60 SCREEN 2,2
70 MAG 2
80 T=0
90 FOR I=0 TO 3:T=T+20
100 IF T<0 THEN T=0
110 IF T>240 THEN T=100
120 S=T:IF T>160 THEN S=T-160
130 SPRITE I,(75,S),I,1
140 NEXT I
150 FOR I=1 TO 100:NEXT I
160 T=T-60
170 GOTO 90
```

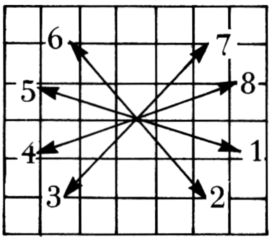
Quatre à quatre

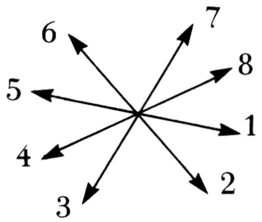
REFERENCES :
Basic Graphique.
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion (un ou deux joueurs).
Difficulté du jeu : difficile.

LE JEU :
Il pourra être pratiqué à un ou plusieurs joueurs. Dans les deux cas, le but est d'arriver à échanger les pions X avec les pions O ; si l'on joue en solitaire, cet échange devra être réalisé avec le moins de coups possible. Les pions sont déplacés alternativement en respectant les règles de déplacement du cavalier aux échecs ; il n'y a pas de prise, et on ne peut passer son tour.

LE PROGRAMME :
La variable alphanumérique A$, en ligne 1Ø, contient la table des coups possibles (déplacement du cavalier).
Les boucles emboîtées I et J (2Ø-6Ø) dessinent le terrain de jeu, et les positions de départ.
La ligne 8Ø permet d'afficher Ø ou X, selon le joueur qui a le trait.
La ligne 1ØØ permet l'affichage du curseur clignotant (ici un +) qui indique sur quelle case l'on se trouve.
La ligne 14Ø regarde si l'on a appuyé sur une touche de 1 à 8, donnant la direction du déplacement.
La ligne 15Ø réimprime la figure qui était sous le curseur.
La ligne 16Ø réalise le contrôle du départ.
Les lignes 17Ø et 18Ø effectuent le déplacement.
Le résultat est imprimé par la ligne 2ØØ, le changement de joueur est réalisé par la ligne 21Ø et un "." est imprimé à l'emplacement de départ.
Le sous-programme 22Ø est utilisé pour les divers contrôles.

POUR JOUER :
On lance le programme par RUN. Le terrain de jeu est affiché, ainsi qu'un symbole permettant d'identifier le joueur qui a le trait. Le joueur peut choisir le pion qu'il désire déplacer en manœuvrant le curseur clignotant grâce aux 4 touches de direction.
Pour déplacer son pion, il suffit, lorsque le curseur est sur ce pion, de taper la direction du déplacement, suivant le code :





Le coup sera refusé si la case de départ n'est pas conforme ou si la case d'arrivée est occupée. Sinon, le coup est réalisé, et le trait passe à l'autre joueur, qui déplace le curseur clignotant jusqu'à l'amener sur l'un de ses pions, et joue de la même façon.
Le joueur gagnant est celui qui réussit à amener le premier tous ses pions dans les cases initialement occupées par l'adversaire. On peut interrompre la partie à tout moment en tapant BREAK.

EXTENSIONS POSSIBLES :
On peut utiliser la couleur, améliorer les contrôles, prévoir un test de fin, utiliser un terrain de jeu différent ; on pourra aussi envisager l'écriture d'un programme permettant au YENO de participer au jeu avec de bonnes chances de l'emporter.

```
     QUATRE A QUATRE

10 CLS:H=1:A$="5665322356"
20 FOR I=1 TO 5:FOR J=1 TO 5
30 CURSOR 10+2*I,10:PRINT CHR$(48+I):C
URSOR 10,10+2*I:PRINT CHR$(48+I):CURSO
R 10+2*J,10+2*I:PRINT "."
40 IF (I<3) AND (J<3) THEN CURSOR 10+2
*J,10+2*I:PRINT "O"
50 IF (I>3) AND (J>3) THEN CURSOR 10+2
*J,10+2*I:PRINT "X"
60 NEXT J:NEXT I
70 X=H:Y=H
80 H$="X":IF H<>0 THEN H$="O"
90 CURSOR 5,22:PRINT "JOUEUR ";H$
100 GOSUB 220:I=J:CURSOR 10+2*X,10+2*Y
:PRINT "+"
110 A=X:B=Y:K$=INKEY$
120 X=X+(K$=CHR$(28))*(X<5)-(K$=CHR$(2
9))*(X>1):Y=Y+(K$=CHR$(31))*(Y<5)-(K$=
CHR$(30))*(Y>1)
130 IF K$="" THEN 150
140 IF (ASC(K$)<57) AND (ASC(K$)>48) T
HEN 160
150 CURSOR 10+2*A,10+2*B:PRINT CHR$(I)
:GOTO 80
160 J=ASC(K$)-48:IF (J>9) OR ((I<>79)
AND (H<>0)) OR ((I<>88) AND(H=0)) THEN
 150
170 X=X+VAL(MID$(A$,J+2,1))-4
180 Y=Y+VAL(MID$(A$,J,1))-4:GOSUB 220:
IF J<>46 THEN 150
190 H$="X":IF H<>0 THEN H$="O"
200 CURSOR 2*X+10,2*Y+10:PRINT H$
210 H=ABS(1-H):I=46:GOTO 150
220 J=VPEEK(40*(10+2*Y)+10+2*X+15362):
RETURN
```

Taquin de lettres

REFERENCES :
Basic Graphique.
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion.
Difficulté du jeu : difficile.

LE JEU :
Qui d'entre nous n'a pas, un jour ou l'autre, eu dans les mains un de ces petits puzzles où l'on essaie de déplacer d'une case à une autre des petits carrés de plastique colorés, jusqu'à l'obtention d'un résultat bien précis ? En son temps le 14-15 de SAM LOYD, a eu autant de succès que notre célèbre RUBIK'S.
Le taquin qui vous est proposé ici utilise 35 lettres, disposées au hasard dans un carré de 6 cases sur 6 cases, et ces lettres peuvent, replacées dans leur ordre initial, vous permettre de reconstituer un proverbe très connu. Une case grise vous permet de déplacer les lettres, et un compteur affiche en permanence le nombre d'essais : arriverez-vous à retrouver le proverbe ?

LE PROGRAMME :
La phrase à reconstituer (qui pourra bien sûr être modifiée) est placée dans une variable R$ en ligne 1Ø.
Le mélange des lettres et l'impression sont faits par la boucle I (2Ø-5Ø).
La ligne 7Ø réalise l'affichage du carré noir, celui-ci est déplacé par la ligne 9Ø en utilisant les 4 flèches.
La ligne 1ØØ réimprime la lettre concernée à l'emplacement du carré noir. Elle comptabilise le nombre d'essais, qui est imprimé grâce à la ligne 7Ø.

POUR JOUER :
On lance le programme par RUN.
La partie sera évidemment d'autant plus difficile que l'on ne connaît pas le proverbe à reconstituer. On peut déplacer les lettres et les faire passer à l'emplacement du carré noir, à l'aide des 4 touches de direction.

EXTENSIONS POSSIBLES :
Le même principe peut être appliqué à toute sorte de taquins (manipulation de nombres, de couleurs, de sons éventuellement). On peut aussi prévoir plusieurs phrases, avec un choix effectué par un générateur aléatoire, etc.

TAQUIN DE LETTRES

```
10 CLS:A$="":R$="ONASOUVENTBESOIND'UNP
LUSPETITQUESOI"
20 FOR I=1 TO 35
30 X=INT(35*RND(1)+1):IF MID$(R$,X,1)=
"-" THEN 30
40 A$=A$+MID$(R$,X,1):R$=LEFT$(R$,X-1)
+"-"+RIGHT$(R$,36-X)
50 CURSOR I-6*INT(I/6)+5,INT(I/6)+5:PR
INT MID$(A$,I,1):NEXT I
60 X=0:Y=0:S=0
70 CURSOR 7,1:PRINT "NOMBRE D'ESSAIS :
";S:CURSOR X+5,Y+5:PRINT CHR$(229)
80 K$=INKEY$:IF K$="" THEN 80
90 I=X:J=Y:X=X+(K$=CHR$(28))*(X<5)-(K$
=CHR$(29))*(X>0):Y=Y+(K$=CHR$(31))*(Y<
5)-(K$=CHR$(30))*(Y>0)
100 C=VPEEK(40*(Y+5)+X+5+15362):CURSOR
 I+5,J+5:PRINT CHR$(C):S=S+1:GOTO 70
```

Cibles

REFERENCES :
Basic Graphique.
Difficulté de programmation : difficile.
Catégorie de jeu : hasard.
Difficulté du jeu : facile.

LE JEU :
Il consiste simplement à tirer sur des cibles, en essayant de marquer le maximum de points ; le nombre de cibles (de 1 à 3) sera choisi en début de partie, et le score réalisé sera affiché en permanence. Si l'on touche le bord d'une cible, la partie s'arrête. On marquera d'autant plus de points que le nombre sur lequel on tombe sera plus grand.

LE PROGRAMME :
Le nombre de cibles est entré en ligne 2Ø. La boucle I ligne 7Ø effectue l'impression des cibles.
La ligne 8Ø affichera le score en permanence.
La boucle est affichée par la ligne 9Ø.
La boucle I (1ØØ-11Ø) réalise l'impression des chiffres contenus dans les cibles, et permet le déplacement de la boule (ligne 11Ø).
La ligne 13Ø regarde si l'on a tiré, et la ligne 14Ø regarde si la boule est encore sur la ligne de départ.
La ligne 15Ø fait progresser la boule d'une case vers la droite, et vérifie si l'on a ou non atteint les cibles ; puis la ligne 16Ø regarde ce que l'on a touché.
La ligne 17Ø incrémente le score (le nombre de points marqués correspond au chiffre affiché).
La ligne 17Ø s'occupe également de la fin de partie.

POUR JOUER :
Lancer le programme par RUN, et taper le nombre de cibles désirées suivi de CR. Les cibles s'affichent, et la partie commence. Votre balle, représentée par un O est affichée en bas à gauche ; vous pouvez la déplacer avec les quatre flèches de direction (↑ , ↓ , →, ←), et vous pouvez tirer à tout moment en tapant Ø.
Le score sera modifié en fonction de votre résultat, et une nouvelle balle apparaît. Il est possible d'arrêter à tout moment en tapant BREAK, sinon le jeu s'arrête au bout d'un certain nombre de tirs.

EXTENSIONS POSSIBLES :
On pourra sans difficulté augmenter le nombre de cibles, modifier la règle d'apparition des nombres dans les cibles pour la rendre moins hasardeuse, s'arranger pour que la balle elle-même soit représentée par un chiffre qui devrait coïncider à l'arrivée avec celui de la cible.

```
10 S=0
20 CLS:BEEP:INPUT "NOMBRES DE CIBLES ?
:";N:CLS
30 P=10:J=25
40 PATTERN C# 128, "FFFF030303030303"
50 PATTERN C# 129, "0303030303030303"
60 PATTERN C# 130, "030303030303FFFF"
70 FOR I=1 TO N:CURSOR J,3*I:PRINT CHR
$(128):CURSOR J,3*I+1:PRINT CHR$(129):
CURSOR J,3*I+2:PRINT CHR$(130):NEXT I
80 K=2:A=21:CURSOR 10,A:PRINT "SCORE :
";S
90 CURSOR K,A:PRINT "0":D=A
100 FOR I=1 TO N:U=INT(P*RND(1)):CURSO
R J-1,3*I+1:PRINT CHR$(48+U)
110 K$=INKEY$:A=A+(K$=CHR$(30))-(K$=CH
R$(31)):NEXT I
120 CURSOR K,D:PRINT " "
130 IF K$="0" THEN 150
140 IF K=2 THEN 90
150 K=K+1:IF K<=22 THEN 90
160 C=VPEEK(40*D+K+1+15362):IF C=32 TH
EN CURSOR 25,21:PRINT "PERDU":END
170 S=S+C-48:P=P-1:IF P<>0 THEN 30
180 END
```

Affichage publicitaire (2)

REFERENCES :
Basic Graphique.
Difficulté de programmation : difficile.
Catégorie de jeu : graphique.
Difficulté du jeu : aucune.

LE JEU :
Ce programme va permettre l'affichage tournant d'un texte en grand format ; il utilise les SPRITES et le générateur de caractères.
On devra redéfinir les caractères que l'on veut afficher car ils apparaissent sur l'écran au moyen de SPRITES magnifiés (lignes 8Ø).
L'exemple donné page suivante a redéfini le E, le N, le O et le Y. La redéfinition débute en 1Ø pour E, et finit en 4Ø pour Y.

LE PROGRAMME :
La ligne 6Ø permettra l'introduction du texte.
Les lignes 1Ø à 4Ø redéfinissent chaque lettre (ici E, N, O, Y).
La ligne 9Ø permet de reboucler indéfiniment sur le défilement.
La boucle J (1ØØ-17Ø) réalise l'affichage lettre par lettre.

POUR JOUER :
On devra d'abord lancer le programme par RUN, puis on entrera le texte à afficher, suivi de CR (en ajoutant 5 INPUT A$).
Le texte commence à défiler sur l'écran ; les lettres apparaissent les unes derrières les autres avec un maximum de 18 sur l'écran.
On peut arrêter le programme à tout moment en tapant BREAK.

EXTENSIONS POSSIBLES :
Il sera possible de reprendre cette idée pour réaliser toute sorte d'affichages fantaisie (couleurs, caractères spéciaux, etc.).

AFFICHAGE PUBLICITAIRE (2)

```
10 PATTERN S# 5, "F88080F08080F800"
20 PATTERN S# 14, "88C8A89888888800"
30 PATTERN S# 15, "F888888888888F800"
40 PATTERN S# 25, "8888502020202000"
50 CLS
60 A$="YENO"
70 SCREEN 2,2
80 MAG 2
90 T=0
100 FOR J=1 TO 13
110 FOR I=0 TO 3:T=T+20
120 S=ASC(MID$(A$,I+1,1))-64
130 SPRITE I,(T,75),S,1
140 NEXT I
150 FOR I=1 TO 100:NEXT I
160 T=T-60
170 NEXT J
180 GOTO 90
```

Anomalie

REFERENCES :
Basic Graphique.
Difficulté de programmation : difficile.
Catégorie de jeu : solitaire.
Difficulté du jeu : moyenne.

LE JEU :
Il repose essentiellement sur une bonne capacité d'observation. Deux grilles de 25 cases chacune sont dessinées sur l'écran. Les lettres de l'alphabet sont affichées en des endroits différents des deux grilles, mais l'une des lettres a été modifiée sur la grille de droite. Le jeu consiste à trouver laquelle, en entrant ses coordonnées dans l'ordre ligne-colonne.

LE PROGRAMME :
La boucle I ligne 1Ø imprime les symboles des deux grilles.
La boucle principale I (ligne 2Ø) imprime les symboles des deux grilles, dans un ordre différent.
La ligne 2Ø choisit le caractère à imprimer.
Le sous-programme (6Ø-8Ø) s'occupe de faire l'impression à un emplacement libre (on l'appelle deux fois pour les deux grilles).
La ligne 3Ø change, de façon aléatoire, l'un des caractères.
Les coordonnées du point essayé sont entrées a cette même ligne, et le contrôle est effectué par la ligne 4Ø qui s'occupe également de l'affichage du résultat.

POUR JOUER :
Après avoir lancé le programme par l'instruction RUN, on obtient l'affichage des deux grilles. On devra alors chercher, par comparaison, quel est le caractère qui a été modifié (sur la grille de droite).
On entre alors son numéro de ligne, et son numéro de colonne, suivi de CR.
Le programme répondra alors BRAVO ou PERDU.

EXTENSIONS POSSIBLES :
Il sera possible d'utiliser des symboles graphiques de diverses couleurs, ce qui pourra simplifier la recherche.
On peut aussi envisager d'utiliser des dessins plus sophistiqués, améliorer le contrôle du résultat, admettre la bonne réponse sur la grille de droite ou sur celle de gauche, introduire un minutage, etc.

ANOMALIE

```
10 CLS:T=5:FOR I=1 TO T:CURSOR 5+2*I,5
:PRINT CHR$(48+I):CURSOR 21+2*I,5:PRIN
T CHR$(48+I):CURSOR 5,5+2*I:PRINT CHR$
(48+I):CURSOR 21,5+2*I:PRINT CHR$(48+I
):NEXT I
20 FOR I=1 TO 25:AL=INT(26*RND(1)+65):
L=1:S=T:GOSUB 60:L=9:GOSUB 60:FOR J=1
TO 100:NEXT J:NEXT I
30 S=0:GOSUB 60:CURSOR 0,20:INPUT "
   LIGNE,COLONNE ? :";I,T:U=2*I-1:V=2*
T+16
40 GOSUB 70:CLS:A$="PERDU":IF AL=C THE
N A$=" BRAVO"
50 CURSOR 10,10:PRINT A$:END
60 U=2*INT(T*RND(1)+1):V=2*INT(S*RND(1
)+L)
70 C=VPEEK(40*(U+5)+V+5+15362):IF (C<>
32) AND (S<>0) THEN 60
80 CURSOR V+5,U+5:PRINT CHR$(AL):RETUR
N
```

Othello

REFERENCES :
Basic Graphique.
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : difficile.

LE JEU :
Devenu un grand classique des jeux sur micro-ordinateur, le jeu d'Othello ou Reversi gagne, même si comme ici il est pratiqué à deux joueurs, à être géré sur ordinateur, avec visualisation : en effet, le jeu consistant essentiellement en des retournements des pions de l'adversaire, ceci peut se révéler, à la longue, très fastidieux, surtout si vous êtes un joueur assidu. Le principe est que le joueur possédant les X doit, pour pouvoir jouer, réussir à encadrer une ou plusieurs des pièces de son adversaire, dans une ou plusieurs directions ; s'il y réussit, les O seront alors retournés pour devenir des X (le retournement pourra être fait horizontalement, verticalement ou suivant les diagonales).
Le joueur ayant les O procédera de la même façon, et ceci jusqu'à ce que le damier soit rempli ou que la personne ne puisse plus jouer (si un joueur ne peut retourner des pions, il doit passer son tour).

LE PROGRAMME :
La boucle P ligne 2Ø dessine le terrain de jeu et la ligne 5Ø affiche un X ou un O, selon le joueur qui commence.
La boucle P (8Ø-14Ø) contrôle la validité du coup (la case est-elle libre : ligne 13Ø), en utilisant une partie de programme qui se trouve aux lignes 16Ø à 19Ø (a-t-on pris un ou plusieurs ennemis en ligne 16Ø et retournement éventuel autorisé par la ligne 17Ø : on refait un balayage dans toutes les directions, si l'encadrement s'est révélé possible).
La ligne 8Ø définit les 8 directions d'exploration.
La ligne 9Ø calcule les déplacements à effectuer pour explorer ces 8 directions.
La ligne 13Ø réalise les déplacements.
La ligne 15Ø assure le changement de joueur, s'il y a eu retournement (contrôlé en ligne 13Ø).

POUR JOUER :
Après RUN, chaque joueur entrera à son tour les coordonnées (ligne, puis colonne, suivi de CR) de la case où il désire jouer ; si le coup est accepté, les pions sont retournés et l'on change de joueur ; si aucun coup n'est possible, on entre Ø en première coordonnée pour le changement de joueur.

EXTENSIONS POSSIBLES :

Avant tout, bien sûr, améliorer l'affichage avec des couleurs, une grille plus aérée.

On pourra aussi imaginer un jeu machine pour le YENO.

Après avoir lancé le programme par RUN, il convient d'entrer la dimension de la grille (8, CR ou 10 ...).

OTHELLO

```
10 CLS:INPUT "DIMENSION : ";A:CLS
20 B=1:FOR P=B TO A:CURSOR 1+2*P,1:PRI
NT P:CURSOR 1,2*P+1:PRINT P:NEXT P
30 CURSOR A+2,A+1:PRINT "X 0"
40 CURSOR A+2,A+3:PRINT "0 X":K=B
50 IF K<>0 THEN CURSOR 15,20:PRINT "JO
UEUR X :":GOTO 70
60 CURSOR 15,20:PRINT "JOUEUR 0 :"
70 CURSOR 2,22:FOR I=1 TO 35:PRINT " "
;:NEXT I:CURSOR2,22:INPUT "LIGNE ?";U:
BEEP:IF U=0 THEN 150
80 CURSOR 15,22:INPUT "COLONNE ?";V:BE
EP:H=PI:FOR P=B TO 8:S=0
90 W=0:T=U+U:R=V+V:F=((P>3) AND (P<7))
-((P<3) OR (P>7)):G=((P>B) AND (P<5))-
(P>5)
100 IF S=1 THEN R$="X"
110 IF S=2 THEN R$="0"
120 CURSOR R+2,T+1:PRINT R$
130 T=T+F+F:R=R+G+G:C=VPEEK(40*(T+1)+R
+4+15362):IF (C=88) OR (C=79) THEN 160

140 NEXT P:IF H<>0 THEN 50
150 K=ABS(K-1):GOTO 50
160 IF ((K<>0) AND (C=79)) OR ((C=88)
AND (K=0)) THEN W=B
170 IF (W=0) OR (S<>0) THEN 140
180 CURSOR R+2,T+1:PRINT R$:IF (W<>0)
AND (((C=79) AND (K<>0)) OR ((C=88) AN
D (K=0))) THEN 130
190 S=B+1-K:H=H*(1-S):GOTO 90
```

Achevé d'imprimer en septembre 1984
sur les presses de l'imprimerie Laballery et Cie
58500 Clamecy
Dépôt légal : septembre 1984

N° d'impression : 409042
N° d'édition : 86595-177-1
ISBN : 2-86595-177-4

Enseigner et distraire est le double objectif de cet ouvrage. Ces 102 programmes de jeux vous entraînent à la découverte du YENO. La classification par niveaux de ces programmes vous permet d'acquérir progressivement des connaissances nouvelles et une plus grande maîtrise du Basic. Pour chaque niveau, la description des jeux proposés est suivie d'un programme abondamment commenté et d'un exemple d'utilisation. Le dernier niveau constitue une illustration claire de l'utilisation du Basic étendu.

DES LIVRES POUR VOTRE ORDINATEUR

ISBN : 2.86595.177.4
110 FF

Couverture GIBRAT

102 PROGRAMMES POUR YENO

PAR JACQUES DECONCHAT

PROGRAMMES

P.S.I.